SHARP

POCKET COMPUTER

MODEL PC-1211

INSTRUCTION MANUAL

MODEL PC-1211 AND PERIPHERALS
LIMITED WARRANTY

Sharp Electronics Corporation warrants each of these products to the original purchaser to be free from defective materials and workmanship. Under this warranty the product will be repaired or replaced, at our option, without charge for parts or labor, with the exception of supplies, such as batteries, ribbons, inked rollers, etc., when returned to a SHARP CONSUMER FACTORY SERVICE CENTER listed in the instruction booklet supplied with your product.

This warranty does not apply to peripheral equipment consisting of cassettes or cassette tape recorders, software programs or appearance items nor to any product whose exterior has been damaged or defaced, nor to any product subjected to misuse, abnormal service or handling, nor to any product altered or repaired by other than a SHARP CONSUMER FACTORY SERVICE CENTER. This warranty does not apply to any product purchased outside the United States, its territories or possessions.

The period of the warranty shall be ninety (90) days on parts and labor from the date of the original purchase.

This warranty entitles the original purchaser to have the warranted parts and labor rendered at no cost for the period of the warranty described above when the unit is carried or shipped prepaid to a SHARP CONSUMER FACTORY SERVICE CENTER together with proof of purchase.

THIS SHALL BE THE EXCLUSIVE WRITTEN WARRANTY OF THE ORIGINAL PURCHASER AND NEITHER THIS WARRANTY NOR ANY OTHER WARRANTY, EXPRESSED OR IMPLIED, SHALL EXTEND BEYOND THE PERIOD OF TIME LISTED ABOVE. IN NO EVENT SHALL SHARP BE LIABLE FOR CONSEQUENTIAL ECONOMIC DAMAGE OR CONSEQUENTIAL DAMAGE TO PROPERTY. SOME STATES DO NOT ALLOW A LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS OR AN EXCLUSION OF CONSEQUENTIAL DAMAGE, SO THE ABOVE LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU. IN ADDITION, THIS WARRANTY GIVES SPECIFIC LEGAL RIGHTS, AND YOU MAY HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

FOR YOUR RECORDS....

For your assistance in reporting this electronic calculator in case of loss or theft, please record below the model number and serial number which are located on the bottom of the unit.
Please retain this information.

Model Number_____ Serial Number _____

Date of Purchase Place of Purchase

_____ _____

OPERATIONAL NOTES

Thank you for your purchase of the SHARP pocket computer, PC-1211.

Since the liquid crystal display is made of glass material, treat the computer with care.
Do not put the "PC-1211" in your back pocket as it may be damaged when you sit down.

To insure trouble-free operation of your SHARP computer, we recommend the following:

1. The computer should be kept in areas free from extreme temperature changes, moisture, and dust.
2. A soft, dry cloth should be used to clean the computer. Do not use solvents or a wet cloth.
3. If the computer will not be operated for an extended period of time, remove the batteries to avoid possible damage caused by battery leakage.
4. If service of your computer is required, use only an authorized SHARP Service Center.
5. Keep this manual for further reference.

It may sometimes cause all of keys inoperative including the CA/BREAK/ON key, when the machine is exposed to a strong external noise. In such a case, press the All Reset switch located on the back panel of the computer to release the situation. However, it will be necessary to load the program again as all of memory and program were reset upon the depression of All Reset switch.
For All Reset switch, see Page 95.

INTRODUCTION

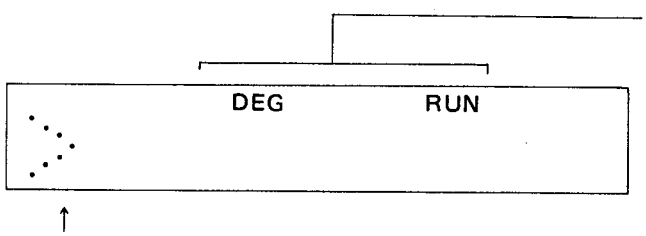
This manual will introduce you to SHARP PC-1211 pocket computer. The PC-1211 is a new powerful computing instrument. It will provide you with formidable computing power in mathematical, scientific, engineering and business calculations.

FEATURES

- Programmable with BASIC language despite its pocket size.
- The 24-digit alphanumeric dot matrix liquid crystal display provides dialogue key operations. This enables the easy use of BASIC language.
- Program capacity 1424 steps, 26 memories with memory safe guard.
- Reservable and definable key systems employed. (See pages 38 and 41)
- Permits, when in connection with an optional cassette interface CE-121, to store programs or data on a cassette magnetic tape or to receive stored programs or data from that tape as occasion demands. (See page 83)

OPERATION

First press the power switch ([ON]).
The machine then gives the following display.



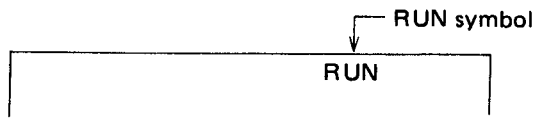
The symbols that show up here vary depending on the machine state just before power-off. (Refer to DISPLAY on page 10.)

↑
Prompt symbol appears.
(This symbol indicates that the PC-1211 is waiting for key inputs.)

The computer is supplied with programs and carries out computations according to the input programs. besides, it can solve problems without being programmed, where they are too simple to require programming, if supplied with necessary data directly through its keyboard.
The former process is called program calculation and the latter one manual calculation – this is also called DIRECT EXECUTION because of direct calculation.

1. Manual calculation (For details, see page 15)

At first make the PC-1211 display the symbol "RUN" at the top of the display of the computer by pressing the [MODE] ([MODE] [MODE]) keys. For the symbol RUN, refer to MODE on page 13.



Now let's enter on practice.

(1) Addition, Subtraction & Playback

Key in the following:

12 [+] 45.6 [−] 32.1 [+] 789 [−] 741 [+] 213

Note that as you key in the "3" in 213, you have exceeded the 24 character capacity of the display. At this point a unique feature called "rolling writer" becomes effective. As each additional step is entered, the display will roll to the left. The data rolled off the screen will be recorded up to 80 steps.

Now press [ENTER] (Do not press the [=] key)

Your answer is 286.5

Now press [►] (playback). You will get back in the display a portion of your original input to check and/or edit. Press [►] each times again to obtain the remainder of your inputs. Editing will be explained in detail in a later section.

If you have placed material in the display and have not used the computer for approximately seven minutes, the computer will go into (APO) "Automatic Power Off" automatically to conserve battery life. (The [OFF] key operation turns the computer off immediately.)

(2) Multiplication, Division

a. Calculate: $841 \times 586 \div 0.12 =$

Key in: 841 [*] 586 [/] .12 [ENTER]

Answer: 4106883.333

As mentioned above, the computer calculates with the aid of BASIC language. Accordingly, the labeling of its multiplication and division keys and others is different from that in normal calculators.

	Normal calculator		PC-1211
Multiplication	[×]	→	[*]
Division	[÷]	→	[/]
Power	[y^x]	→	[SHFT] [∧]

b. Calculate: $427 + 54 \times 32 \div 7 - 39 \times 2 =$

Key in: 427 [+] 54 [*] 32 [/] 7 [−] 39 [*] 2 [ENTER]

Answer: 595.8571429

Note that multiplication and division have priority to addition and subtraction.

(3) Scientific functions

a. Calculate: $\sin 30^\circ + \cos 40^\circ =$

You must set the angular mode to degree, at first.

Press [D] [E] [G] [R] [E] [E] [ENTER] in this sequence.

The symbol "DEG" will be displayed at the top of the display. (Refer to page 10)

Key in: [S] [I] [N] [3] [0] [+] [C] [O] [S] [4] [0] [ENTER]

Answer: 1.266044443

b. Calculate: $1.2 \times 10^{20} \times 1.5 \times 10^{-5} =$

Key in: 1.2 [Exp] 20 [*] 1.5 [Exp] [−] 5 [ENTER]

Answer: 1.8E 15 (1.8×10^{15})

(4) "Expression" Correction and editing (See page 23, Editing expressions)

a. To clear the computer:

Press the red key in upper right corner of the computer marked "CL" and perform the calculation from original step.

b. Correction

The flickering cursor is used to correct and edit expressions.

i) Before pressing the [ENTER] :

Key in: 12 [+] 3 [*] 2 [−] SIN 30 (DEG mode)

If you wish to change the multiplier from 2 to 4, press the [◄] key 7 times until the cursor covers the 2; now press 4; press [ENTER] (If is not necessary to move the cursor back)

Answer: 23.5

ii) After pressing the [ENTER] :

Key in: 12 [+] 3 [*] 2 [−] SIN 30 [ENTER] (DEG mode)

Answer: 17.5 Press [◄] or [►] (Playback)

The cursor is flashing over the (1). If you wish to change the multiplier from 2 to 4, press the [►] key 5 times until it covers the 2; now press 4; press [ENTER] (If is not necessary to move the cursor back)

Answer: 23.5

To change SIN 30 to COS 45, move the cursor to over the SIN and press COS 45 and [ENTER]

Answer: 23.29289322

c. Deletion

Key in: 12 [+] 45 [+] 78

To change the 45 to 5, press the cursor key ([◄]) 5 times until it covers the 4, press [SHFT] [DEL] . The 4 will disappear and all remaining material in the display will shift automatically to the left.

d. Insertion

To insert material place the cursor over the item which will follow the insertion. Multiple insertions may be made.

Key in: 2 ([SHFT] π [+] 4) [ENTER] Display indicates error. A multiplication sign must be placed between the 2 and (. Press [►] or [◄] . The cursor is flashing over the (and is indicating error position. Press [SHFT] [INS] . An opening occurs on the display and all material following the insertion automatically shifts to the right.

Press [*] and [ENTER]

Answer: 14.28318531

2. Program calculation (For details, see page 28.)

For your easier understanding, we will discuss the program calculation taking the case of Law of Cosines.

a. Programming

First put the PC-1211 in the program mode (PRO).

Press the [MODE] key until the symbol "PRO" is displayed.

Second clear the program memory.

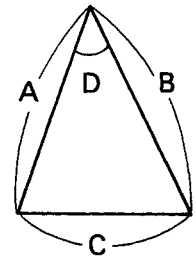
To clear the program memory,

Press [N] [E] [W] [ENTER] .

Example: Solves for C knowing A, B and D
The basic equation is

$$C = \sqrt{A^2 + B^2 - 2AB \cos D}$$

A = 3
B = 4
D = 60°, 45°



Set the angular mode to degree.

Press [D] [E] [G] [R] [E] [E] [ENTER]

(When the symbol "DEG" is already on the display, this operation is not needed.)

Key in:

See SECOND FUNCTION on page 12.

10 [I] [N] [P] [U] [T] A [SHFT] [,] B [SHFT] [,] D

Press the [ENTER] key. The computer then offers the following display automatically.

```
10 : INPUT   A, B, D
```

(The above operation is BASIC-language-based inputting.)

10 : Line number, INPUT : Key word (Statement)
A · · · · · · · , D : Operand, [ENTER] : Commands to end one line.

20 [C] [=] [√] (A * A + B * B − 2 * A * B * COS D) [ENTER]
30 [P] [R] [I] [N] [T] C [ENTER]
40 [E] [N] [D] [ENTER]

b. Execution of the program

Set the PC-1211 at the RUN mode by pressing the [MODE] key.
Next command the computer to start the execution of the program.

Press [R] [U] [N] [ENTER]

This will cause the following display.

```
?          DEG        RUN
```

↑— This display indicates that the computer is requiring a numerical input.

Press	Display	Note
3 [ENTER]	?	A
4 [ENTER]	?	B
60 [ENTER]	3.605551275	Answer

When you want to calculate for another numerical values with aid of the same program, press the [R] [U] [N] and [ENTER] key again.
(This key operation is automatically made, if programmed in advance. But its description is too complicated to give here.)

Press	Display	Note
RUN [ENTER]	?	
3 [ENTER]	?	
4 [ENTER]	?	
45 [ENTER]	2.833626167	Answer

3. Cassette Magnetic Tape (CMT) (Optional feature); for details, see page 83.)

We will brief hereunder the method of storing the program for Low of Cosines – mentioned in 2 above – on a CMT or of recording that program, in reverse, from the CMT to the PC-1211.

(1) Prepare an optional cassette interface CE-121.

(2) For connection of the PC-1211, CE-121 and a tape recorder, consult page 83.

(3) a. **PC-1211 ⇒ CMT (Program storing)**

Press [C][S][A][V][E] [SHFT] [" "] A2 [SHFT] [" "]

FILE NAME (The file name can be determined by yourself. Here, A2 is the file name.)

b. Press the [ENTER] key. The cassette tape recorder begins to rotate and nothing but symbol "RUN" shows up at the top of the display, thereby indicating that a certain program is being written on the tape. On the completion of writing, the PC-1211 pips and concurrently stops the tape movement with the prompt symbol (>) on the display.

(4) **CMT ⇒ PC-1211 (Program loading)**

a. To proceed to the program loading directly after the above operation, press the STOP button on the tape recorder to free it from recording phase and then disconnect the black plug from the remote terminal jack on the tape recorder. Thereafter, rewind the cassette tape. (Rewind the tape to the beginning of its recorded area. Search that beginning at your own guess or by the aid of a tape counter if available.)
own guess or by the aid of a tape counter if available.)

b. Insert the black plug into the remote terminal jack again.

c. Press NEW [ENTER] . (In this step, clear the memory in advance on purpose to test if the program is perfectly loaded on the PC-1211.)

d. Press CLOAD [SHFT] [" "] A2 [SHFT] [" "]
Press the [ENTER] key. The tape then rotates to load the PC-1211 with the information recorded on the tape.
On the completion of loading appears the prompt symbol (>) on the display.
To check that the loading is perfectly made, press the [↑] key. At each press on this key the line numbers of the loaded program are displayed in sequence from the final one to the first.
In case error code "5" appears during loading, repeat the operation of (4) above from the beginning. (For details, see page 87)

(5) Let's execute the program transferred from the CMT to the PC-1211.

a. Set the PC-1211 at the RUN mode.

b. Press RUN [ENTER] . The symbol "?" will be displayed for requring a numerical input.

Press	Display
3 [ENTER]	?
4 [ENTER]	?
60 [ENTER]	3.605551275

This shows that the PC-1211 is certainly loaded with the program from the CMT.

INDEX

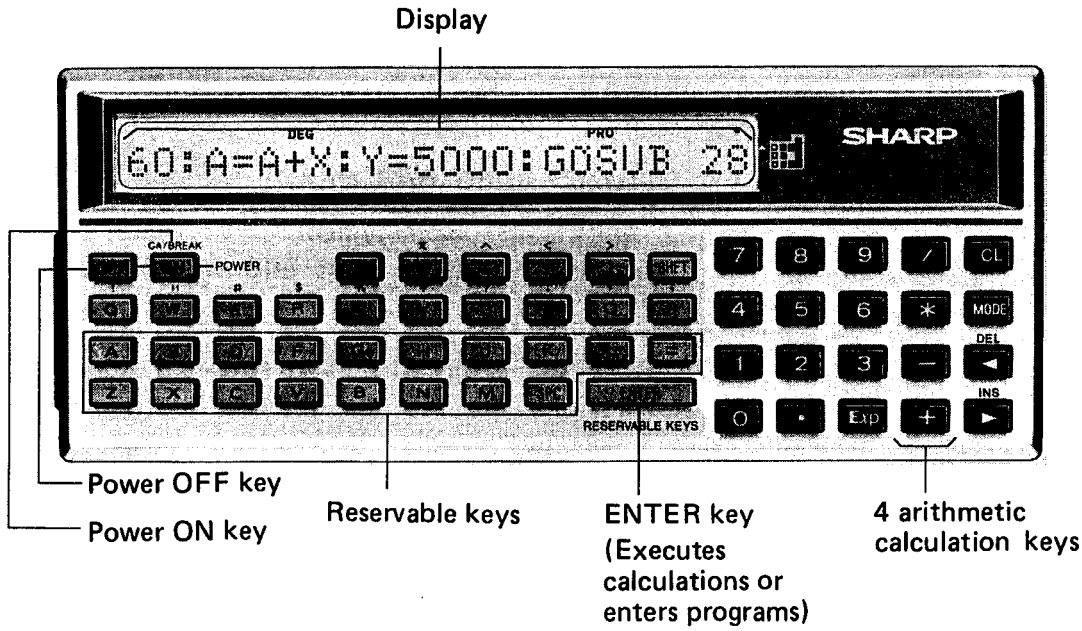
Page

- OPERATIONAL NOTES 1
- INTRODUCTION 2
- THE KEYBOARD 9
- DISPLAY 10
- SECOND FUNCTION 12
- MODE 13
- READING-IN SYSTEM OF DATA 13
- COMPUTATION RANGE 14
- DISPLAY SYSTEM 14
- MANUAL CALCULATIONS 15
  1. What is a manual calculation? 15
  2. For arithmetic calculations 15
  3. Power calculation 17
  4. Calculation with parentheses 17
  5. Scientific functions 18
  6. Logic functions 21
  7. Calculations using memories 22
  8. Successive designation of expressions in manual calculation 22
  9. PLAYBACK function 23
  10. Editing expressions 23
  11. Priority of calculations 26
- PROGRAMMING 28
  1. What is a program calculation? 28
  2. Wirting programs 30
  3. Checking stored programs 32
  4. Program correction 33
  5. Executing programs 35
  6. Debugging programs 37
  7. Defined program 38
- RESERVABLE KEY 41
  1. Reserve for reservable keys 41
  2. Use of reservable keys 42
  3. Checking reserve programs 44
  4. Correction of reserve programs 44
  5. Deleting reserve programs 45
  6. Configuration of reserve programs 45
- VARIABLES 46
  1. What is a variable? 46
  2. Specifying variables 47
  3. Inputting to variables 49
  4. Recalling the contents of variables 50
- PROGRAM STATEMENTS 52
  1. LET statement 52
  2. INPUT statement 53
  3. PRINT statement 55
  4. PAUSE statement 58
  5. USING statement 58
  6. GOTO statement 60
  7. IF statement 61
  8. GOSUB statement, RETURN statement 63

9. FOR statement, NEXT statement 65
10. STOP statement 69
11. END statement 69
12. BEEP statement 69
13. CLEAR statement 69
14. DEGREE, RADIAN, GRAD statements 70
15. AREAD statement 70
16. REM statement 71
- COMMAND STATEMENTS 71
1. RUN command 71
2. DEBUG command 72
3. CONT command 72
4. LIST command 73
5. NEW command 74
6. MEM command 74
- ERROR 75
- MAGNETIC TAPE CONTROL INSTRUCTIONS 77
1. CSAVE (Cassette Save) statement 77
2. CLOAD (Cassette Load) statement 78
3. CLOAD? (Cassette Load?) statement 78
4. CHAIN statement 79
5. PRINT # (Print cross-hatch) statement 81
6. INPUT # (Input cross-hatch) statement 82
- CONNECTION OF CASSETTE INTERFACE (CE-121) 83
1. Battery replacement 83
2. Connecting the PC-1211 with the CE-121 84
3. Connecting the CE-121 with a tape recorder 85
- MAGNETIC TAPE CONTROL INSTRUCTIONS AND TAPE RECORDER OPERATING PROCEDURES 86
1. Adjustments of the tape recorder and hints during adjustments 86
2. Recording to the magnetic tape (CSAVE and PRINT# command) 87
3. Collation (CLOAD? command) 88
4. Transfer from the tape (CLOAD, CHAIN, and INPUT# commands) 89
- FUNCTIONS OF KEYS 91
- BATTERY REPLACEMENT 95
- SPECIFICATIONS 96
- LIST OF FUNCTIONS AND STATEMENTS 97
1. Functions 97
2. Statements 98

Name label
Write your name on the attached name label and stick it on the back of the computer.

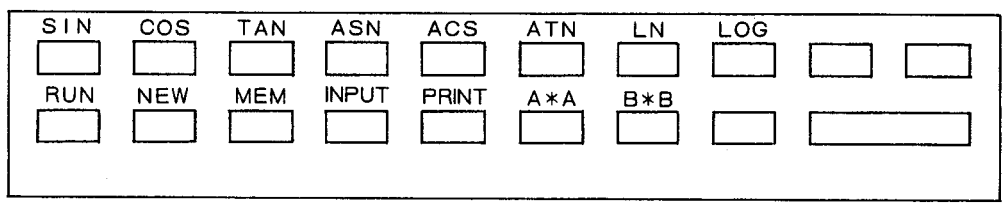
THE KEYBOARD



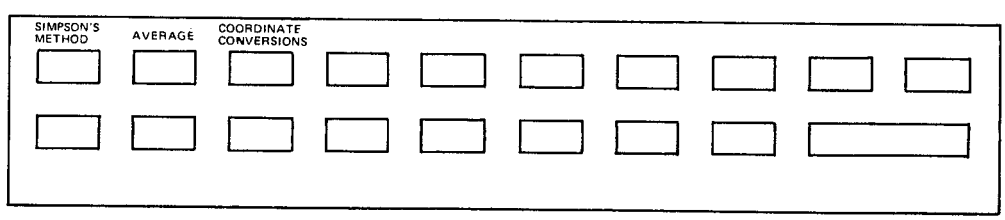
Templates:

Two templates are supplied with the machine. Write on each template the labels of key operations assigned to the reservable keys or those of defined programs assigned to the definition keys, and fit them in the keyboard.

Example: Assignment of reserves to keys (For reservable key: refer to page 41.)



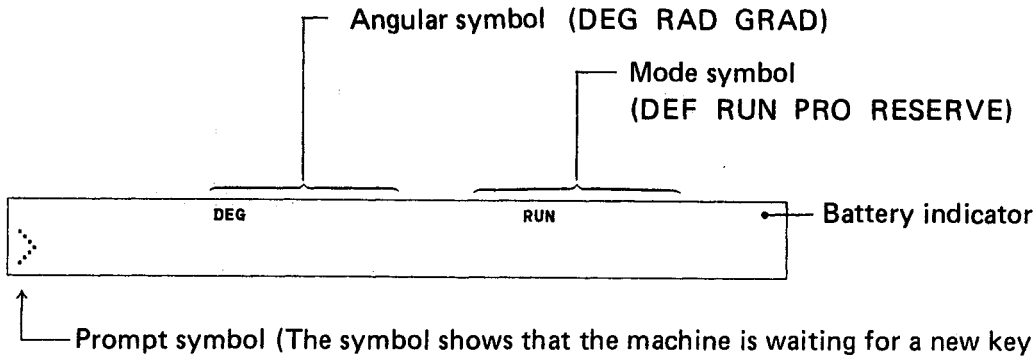
Example: Assignment of defined programs to keys (For defined program, see page 38.)



DISPLAY

The PC-1211 has a 24-digit dot matrix liquid crystal display.

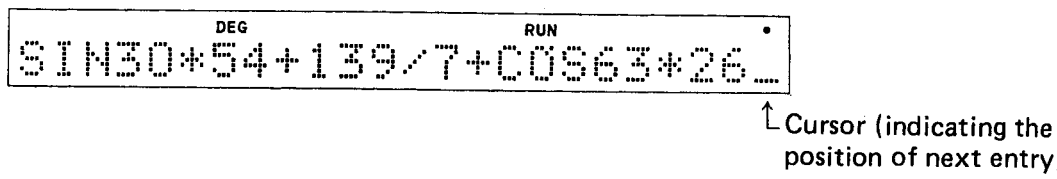
(1) A display that occurs when the power is on or when the mode is changed.


Angular symbol (DEG RAD GRAD)
Mode symbol
(DEF RUN PRO RESERVE)
DEG
RUN
Battery indicator
Prompt symbol (The symbol shows that the machine is waiting for a new key input.)

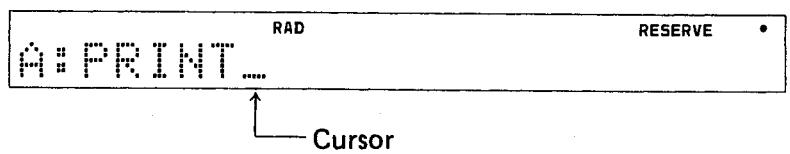
- When the power is on immediately after battery replacement, the prompt symbol, DEG and RUN appear.
- When switched on except immediately after battery replacement, the computer displayed the prompt symbol, as well as an angular symbol and mode symbol which had been on the display just before the power supply was cut through the [OFF] key or the automatic power-off function.
- When you change the machine mode by pressing the [MODE] key, the PC-1211 displays the prompt symbol, the then existing angular symbol and a mode symbol which corresponds to the mode newly chosen.

(2) A display that occurs when you input an "expression" or others through the keys.

(1) RUN mode:



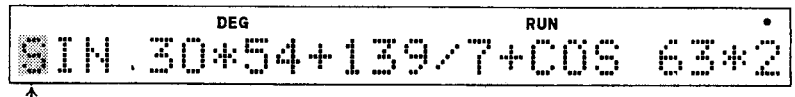
(2) RESERVE mode:



- If a new key operation causes the display to exceed 24 digits, the previous display is shifted to the left to provide a space to display the new input.

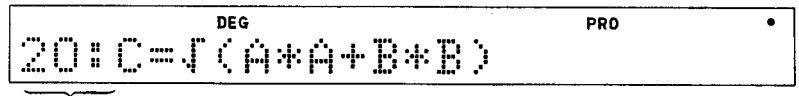
(3) A display of recalled information

(1) RUN mode:



Cursor (If the step indicated by the cursor is filled with an instruction, a symbol of that step and all of dots contained in one-digit display of that step are alternately displayed as cursor display.)

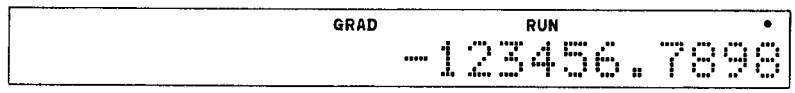
(2) PRO mode:



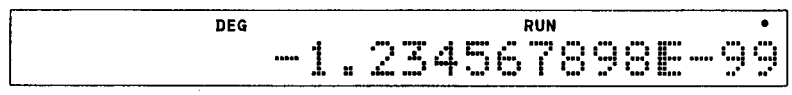
Line number (Displays the line of program. Refer to page 31)

(4) Calculation result

(1) Fixed decimal point system:



(2) Floating decimal point system (Scientific notation)

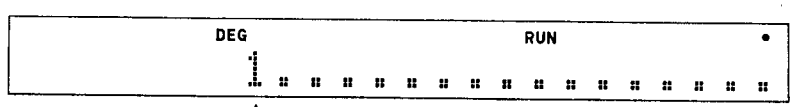


Mantissa Exponent

- Numerical values such as calculation results are all displayed as being shifted to the right.

(5) Error condition

(1) Manual calculation



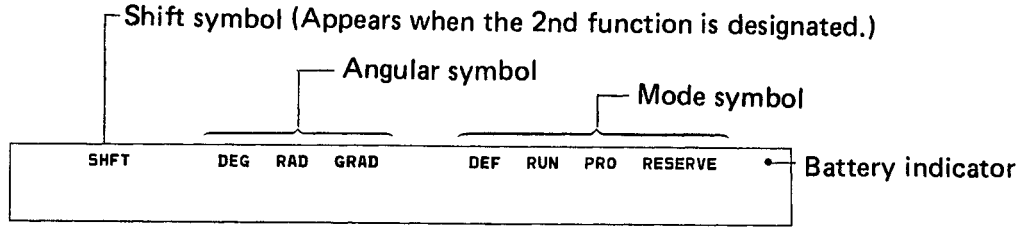
Error code

(2) Program calculation

Error code

Line number (Displays the line number for which an error is detected.)

(6) Symbol



Angular symbols

DEG: Appears when DEGree mode is set.
RAD: Appears when RADian mode is set.
GRAD: Appears when GRAD mode is set.

Mode symbols

DEF: Appears when the DEFinable mode is set.
RUN: Appears when the RUN mode is set.
PRO: Appears when the PROgram mode is set.
RESERVE: Appears when the RESERVE mode is set.

- Battery indicator
  The battery indicator is a grey dot located in the upper right corner of the display. When this dot is not on, the batteries must be replaced.

SECOND FUNCTION

The yellow key of the computer marked "SHFT" must be used to designate the material appearing in mustard above each key. When this key is pressed, the designation "SHFT" will appear in the upper part of the display. If you press this key in error, press it a second time and the "SHFT" designation will disappear.

Example: [SHFT] [$\sqrt{\ }$ (Λ)] → "Λ" is entered.

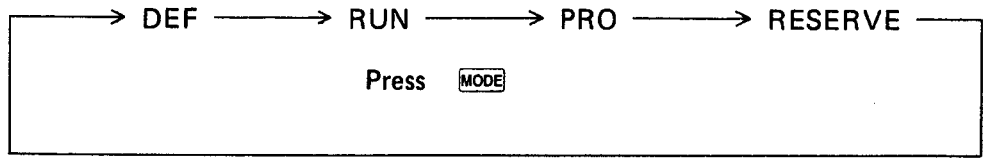
> In this instruction manual, the key operation of second function is shown as follows;
> Example: [SHFT] [$\sqrt{\ }$ (Λ)] → [SHFT] [Λ]

MODE

The PC-1211 has four modes: definable, run, program and reserve program, each mode can be set by pressing the [MODE] key located at the right side of the computer.

Definable mode (DEF):	Puts the machine in the defined program execution mode. Perform defined program calculations in this mode.
Run mode (RUN):	Puts the machine in the calculation execution mode. Perform program or manual calculations in this mode.
Program mode (PRO):	Puts the machine in the program writing mode. Enter programs in this mode.
Reserve program mode (RESERVE):	Puts the machine in the reserve program writing mode. Enter reserve programs in this mode.

The [MODE] key changes the mode.



READING-IN SYSTEM OF DATA

To input a number in the machine, operate the [+] or [−] key first for inputting a sign, and then a numeric key or the decimal point key. (The operation of the [+] key can be omitted, however.) To input a number in the scientific notation system ($A \times 10^B$), input the mantissa, press the [Exp] key and input the exponent.

Ex. -12.345 → [−] [1] [2] [•] [3] [4] [5]

6.7×10^8 → [6] [•] [7] [Exp] [8]

-9.12×10^{-34} → [−] [9] [•] [1] [2] [Exp] [−] [3] [4]

For the input data of mantissa only its upper 10 digits are effective, but the weight of the data is retained as much as the whole input data. For a number smaller than 1 but larger than −1 the data is also retained to a maximum of 10 digits, and its weight is retained as equal to that of input data.

Ex. 1234567898765 → equal to $1.234567898 \times 10^{12}$

9.87654321234 → equal to 9.876543212

0.0000000002345678 → equal to 2.345678×10^{-10}

0.00001234567 [Exp] 24 → equal to 1.234567×10^{19}

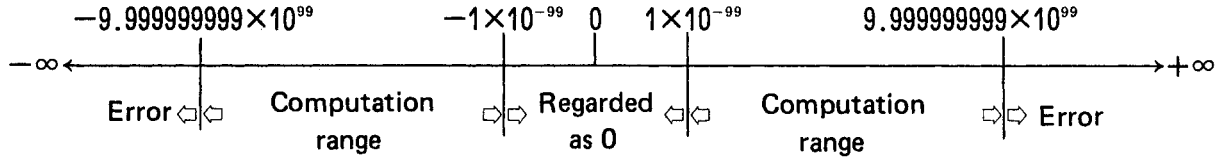
For the exponent its last 2 digits are effective.

Ex. 3 [Exp] 123 → equal to 3×10^{23}

4 [Exp] [−] 3210 → equal to 4×10^{-10}

COMPUTATION RANGE

The computing range is $-9.999999999 \times 10^{99}$ to -1×10^{-99}, 0 and 1×10^{-99} to $9.999999999 \times 10^{99}$.
In case of a number beyond above range, the number is taken to be an overflow error or 0. (See the illustration below.)



DISPLAY SYSTEM

This machine displays a number in the fixed decimal point system or the floating decimal point system (scientific notation system). Numbers in the program calculation are displayed according to the designated format, but in the manual calculation, numbers within the following range are displayed in the fixed decimal point system, in principle, and other numbers are displayed in the floating decimal point system.

Range of numbers displayed in the fixed decimal point system:

$$-9999999999 \leqq x \leqq -1 \times 10^{-9}$$
$$x = 0$$
$$1 \times 10^{-9} \leqq x \leqq 9999999999$$

- For a number within the range shown above that can not be displayed in the fixed decimal point system, the display system is automatically changed over to the floating decimal point system.

  Ex. $0.000123456 \vdots 78 \rightarrow 1.2345678 \times 10^{-4}$

  A calculation result is displayed in the fixed decimal point system or the floating decimal point system, but it is stored in the memory in the form of

  $$A \times 10^{B} \quad (1 \leqq |A| \leqq 9.999999999,\ -99 \leqq B \leqq 99)$$

  or as 0.

MANUAL CALCULATIONS

1. What is a manual calculation?

The PC-1211 is usually programmed in the PRO mode and executes given programs in the RUN or DEF mode. For problems that do not need programming, however, the computer permits to input data necessary for the solution of them in the RUN (or DEF) mode and processes those data direct. (Direct Execution) This method is called manual calculation.

General form **(Expression)** [ENTER]

Example: 5 [*] 4 [ENTER]

Input an expression and press the [ENTER] key. Then the computer will show the answer of the expression.

- **Manual calculations given in the following examples are executed in the RUN mode. Set the machine to the RUN mode by pressing the [MODE] key. (The symbol "RUN" then appears on the display.)**

An Expression is composed of the following instructions:

- Constant ..... 0 ~ 9, ·, π, Exp
- Sign ..... +, −
- Arithmetic operator ..... +, −, * (Multiplication), / (Division), ∧ (Power)
- Logic operator ..... =, >, <, >=, <=, <>
- Functions ..... SIN, COS, TAN, ASN, ACS, ATN, LN, LOG, EXP, DEG, DMS, INT, ABS, SGN, √
- Parenthesis ..... (,) ·
- Memories ..... A ~ Z, A( )

An "Expression" can be made by combining these instructions according to a mathematical formula.
A mathematical formula is defined as "Expression", even if it is composed only of constants or memories. (Ex. 12, π, A etc).

2. For arithmetic calculations

(1) Addition and subtraction

Ex. 7 − 9 + 14 =
−4.2 + 5 − 12.3 =

Operation	Display	Note
RUN mode		
[CL] 7 [−] 9 [+] 14	7−9+14_	Expression
[ENTER]	12.	Ans.
[CL] [−] 4.2 [+] 5 [−] 12.3	−4. 2+5−12. 3_	Expression
[ENTER]	−11. 5	Ans.

(2) Multiplication and division

Ex. 12∗24/5= (12×24÷5=)
27E3∗4/12= (27×10³×4÷12=)

BASIC language ∗ : Multiplication
/ : Division
E : Exponent

Operation	Display	Note
CL 12 ∗ 24 / 5	12∗24/5_	Expression
ENTER	57. 6	Ans.
CL 27 Exp 3 ∗ 4 / 12	27E3∗4/12_	Expression
ENTER	9000.	Ans.

(3) Mixed calculation

Ex. 54+24.3∗16.49/3.4−37.4=

Operation	Display	Note
CL 54 + 24.3 ∗ 16.49	54+24. 3∗16. 49_	
/ 3.4 − 37.4	54+24. 3∗16. 49/3. 4−37. 4_	
ENTER	134. 455	

Note that multiplication and division have priority to addition and subtraction.

(4) When utilizing a displayed result in the subsequent calculation.

In each of the above examples, the CL key is pressed first. This operation is intended to clear the preceding operations or the results of calculations.

If this operation is not made, the previous calculation result may be incorporated into a new expression.

Example ① 3 + 4 =
② −5 + 6 =

The result of (1) is incorporated into the expression (2), thus calculation 3 + 4 − 5 + 6 = is carried out.

Operation	Display	Note
CL 3 + 4 ENTER	7.	The result of ①
→ −	7. −_	The result of ① , (7) is incorporated as data into the expression (2).
5 + 6	7. −5+6_	
ENTER	8.	

If you press keys such as + , − , ∗ , / just before inputting an expression, the preceding calculation result is incorporated as data into that

3. Power calculation

Ex. $4 \wedge 3 =$ $(4^3=)$

$3 \wedge 3.2 * 4 \wedge 2.4 =$ $(3^{3.2} \times 4^{2.4}=)$

$4 \wedge 3 \wedge 2 =$ $(4^{3^2}=)$

Operation	Display	Note
[CL] 4 [SHFT] [∧] 3	4^3_	Expression
[ENTER]	64.	Ans.
3 [SHFT] [∧] 3.2 [*]	3^3.2*_	Expression
4 [SHFT] [∧] 2.4	3^3.2*4^2.4_	
[ENTER]	936.9836103	Ans.
4 [SHFT] [∧] 3 [SHFT] [∧] 2	4^3^2_	Expression
[ENTER]	262144.	Ans.

Note that power calculation has priority to four arithmetic calculations.

4. Calculation with parentheses

This calculation can be carried out by using the [(] and [)] keys in the same manner as you use parentheses generally employed in mathematical formula.

Ex. $(72+9)/4*\{21*\lfloor 68/(7-3)+2 \rfloor\}=$

Operation	Display	Note
[CL] [(] 72 [+] 9 [)] [/] 4 [*]	(72+9)/4*_	
[(] 21 [*] [(] 68 [/] [(] 7	(72+9)/4*(21*(68/(7_	
[−] 3 [)] [+] 2 [)] [)]	+9)/4*(21*(68/(7−3)+2))_	
[ENTER]	8079.75	Ans.

The use of parentheses gives a quite different meaning as follows. Use due care when using instruction " / " or a function, in particular.

$A+B/C \rightarrow A+\frac{B}{C}$ $\qquad \sqrt{\ }A+B \rightarrow \sqrt{A}+B$

$(A+B)/C \rightarrow \frac{A+B}{C}$ $\qquad \sqrt{\ }(A+B) \rightarrow \sqrt{A+B}$

$A/C*D \rightarrow \frac{AD}{C}$ $\qquad \sqrt{\ }A*B \rightarrow B\sqrt{A}$

$A/(C*D) \rightarrow \frac{A}{CD}$ $\qquad \sqrt{\ }(A*B) \rightarrow \sqrt{AB}$

$A/B/C \rightarrow \frac{\frac{A}{B}}{C}=\frac{A}{BC}$ $\qquad A*B+C \rightarrow AB+C$

$A/(B/C) \rightarrow \frac{A}{\frac{B}{C}}=\frac{AC}{B}$ $\qquad A*(B+C) \rightarrow A(B+C)$

5. Scientific functions

The machine permits a function to be calculated in the same procedure as you calculate it according to mathematic formula.

- When performing a functional calculation of a constant or memory, input it following a function as SIN 30 or SIN A. In other cases be sure to input it following a function as parenthesized like LN (A * B) or SIN (π/2).

① Angle mode

The angular mode is designated by the followings:

Degree mode:	[D][E][G][R][E][E][ENTER]	("DEG" will appear at the top of the display.)
Radian mode:	[R][A][D][I][A][N][ENTER]	("RAD" will appear)
Grad mode:	[G][R][A][D][ENTER]	("GRAD" will appear)

② Trigonometric functions (SIN, COS, TAN)

Ex. SIN 30= (sin30=) Set the angular mode to "DEG".
COS (π/4)= ($\cos\frac{\pi}{4}$=) RAD mode
TAN 150= (tan150=) GRAD mode

Operation	Display	Note
DEG ([D][E][G][R][E][E][ENTER])		"DEG"
[CL] [S][I][N] 30	SIN30_	
[ENTER]	0. 5	SIN 30°
RAD		
[C][O][S][(][SHFT][π]	COS (π _	
[/] 4 [)]	COS (π ／4) _	
[ENTER]	7. 071067812E−01	$\cos\frac{\pi}{4}$(rad)
GRAD [T][A][N] 150	TAN150_	
[ENTER]	−1.	TAN 150^g

(3) Inverse trigonometric functions (ASN, ACS, ATN)

Ex. ASN−0.5= ($\sin^{-1}$(−0.5)=) ASN: Arcsine
ACS (−0.5+0.1)= ($\cos^{-1}$(−0.5+0.1)=) ACN: Arccosine
ATN (7/3)= ($\tan^{-1}\frac{7}{3}$=) ATN: Arctangent

Operation	Display	Note
DEG[CL]		
[A][S][N][−].5	ASN−. 5 _	
[ENTER]	−30.	(°)
RAD[A][C][S][(][−].5	ACS (−. 5_	
[+].1 [)]	ACS (−. 5+. 1) _	
[ENTER]	1. 982313173	(rad)
GRAD [A][T][N][(] 7	ATN (7 _	
[/][3][)]	ATN (7／3) _	
[ENTER]	74. 22378832	(^{g})

(4) Logarithmic functions (LN, LOG)

Ex. LN7.4= (ln7.4=)
LOG100= (log100=)

[Note: $\ln X = \log_e X$: Natural logarithm
$\log X = \log_{10} X$: Common logarithm]

Operation	Display	Note
CL L N 7.4	LN7. 4_	
ENTER	2. 00148	Ans.
L O G 100	LOG100_	
ENTER	2.	Ans.

(5) Exponential functions (EXP)

Ex. EXP−13.6= ($e^{-13.6}$) [Note: EXP is LN of anti-logarithm]

Operation	Display	Note
CL E X P − 13.6	EXP −13. 6_	
ENTER	1.24049508E−06	Ans.

(6) Roots

Ex. $\sqrt{\ }73=$ ($\sqrt{73}=$)
$\sqrt{\ }\sqrt{\ }256=$ ($\sqrt{\sqrt{256}}=\sqrt[4]{256}=$)
$\sqrt{\ }(3*3+4*4)=$ ($\sqrt{3^2+4^2}=$)

Operation	Display	Note
CL √ 73	√73_	
ENTER	8. 544003745	Ans.
√ √ 256	√√256_	
ENTER	4.	Ans.
√ (3 *	√(3*_	
3 + 4 * 4)	√(3*3+4*4) _	
ENTER	5.	Ans.

(7) Angle conversions (DMS, DEG)

DMS: Decimal degrees → Degrees/minutes/seconds
When converting decimal degrees to the equivalent degrees/minutes/seconds, the answer is broken down: integer portion = degrees; 1st and 2nd decimal digits = minutes; 3rd and 4th digits = seconds; and the 5th through end decimal digits are decimal degrees.

DEG: Degrees/minutes/seconds → Decimal degrees
To convert an angle given as degrees/minutes/seconds to its decimal equivalent, it must be entered as integer and decimal respectively.

Ex. Convert 15.4125° to its degree/minute/second equivalent.
Convert 15°24'45" to its decimal equivalent.

Operation	Display	Note
CL DMS 15.4125	DMS15. 4125_	
ENTER	15. 2445	15°24′45″
DEG 15.2445	DEG15. 2445_	
ENTER	15. 4125	15.4125°

(8) Integer (INT)

The function converts numerical values such as answers of expressions to the largest integers not larger than those values: 12.34 to 12. or −2.45 to −3., for example.

Ex. INT (65/3)=
INT (−0.3)=

Operation	Display	Note
INT (65 / 3)	INT (65/3) _	
ENTER	21.	
INT − .3	INT −. 3 _	
ENTER	−1.	

(9) Sign function (SGN)

The function takes the following values for numerical value X.

+1 if X > 0
0 if X = 0
−1 if X < 0

Ex. SGN (5−9)=

Operation	Display	Note
SGN (5 − 9)	SGN (5−9) _	
ENTER	−1.	

(10) Absolute value (ABS)

The function finds out the absolute value | X | of a numerical value X.

Ex. ABS (5−9)= (|5−9|=)

Operation	Display	Note
ABS (5 − 9)	ABS (5−9) _	
ENTER	4.	

6. Logic functions

These functions take 1 when an expression composed using logic operators (=, >, <, >=, <= and <>) is true, and 0 when false. In other words, $x \circ y$ ($\circ$ is a logic operator) takes 1 or 0 depending on the relations of x and y.

Logic operator	
=※	1 if $x = y$. 0 if $x \neq y$.
>	1 if $x > y$ 0 if $x \leqq y$
<	1 if $x < y$ 0 if $x \geqq y$
>=	1 if $x \geqq y$ 0 if $x < y$
<=	1 if $x \leqq y$ 0 if $x > y$
<>	1 if $x \neq y$ 0 if $x = y$

Note: < > has the same meaning as ≠.

※ The logic functions using = are, if set up in the form of [memory] = ⟨ expression ⟩ like A = ⟨ expression ⟩, B = ⟨ expression ⟩, for example, not judged to be logic functions but inputs to the memory (substitutional statement). To prevent this, you must compute them in the form of ⟨ expression ⟩ = [memory]. This is not true of relative equations in IF statements, however.

Ex. (5+8)>(3*4)=

(24/5)<=(2.4*2)=

Operation	Display	Note
(5 + 8) SHFT >	(5+8) > _	
(3 * 4)	(5+8) > (3*4) _	
ENTER	1.	Ans.
(24 / 5) SHFT < =	(24/5) <=_	
(2.4 * 2)	(24/5) <= (2. 4*2) _	
ENTER	1.	Ans.

= Reference =

Logical sum (OR) and product (AND) can be formed by the following composition of logical computations.

① Logical sum (logical computation) + (logical computation)

Example: (A < 0) + (A > 8) 1 is taken if A is smaller than 0 or larger than 8.

(B > 0) + (C > 0) 1 (2) is taken if B or C is larger than 0 and both B and C are larger than 0.

② Logical product (logical computation) * (logical computation)

Example: (B > 1) * (B < 6) 1 is taken if B is larger than 1 and smaller than 6.

7. Calculations using memories

The PC-1211 has two types of data memories: fixed memory – 26 pieces in all – and flexible memory. In this paragraph we will describe the method of calculation using fixed memories. (For details, refer to page 46.)

(1) Specifying memories ①

The fixed memories are given labels A through Z, each being specified by the [A] through [Z] keys.

Example: When the memory A is loaded with 4 and the memory B with 5.

[A] [+] [B] [*] [B] [−] 12 [ENTER] → 17.

[√] [(] [A] [+] [B] [)] [ENTER] → 3.

([A] [=] 4 [ENTER] / [B] [=] 5 [ENTER])

(2) Specifying memories ②

The fixed memories A through Z are numberd from 1 through 26, each being specified in the form of A ( ). (When you specifiy the memories by this method, those are called dimension memories, in particular.)

Example: [A] [(] 2 [)] → Memory A (2), namely memory B, is specified.

[A] [(] 2 [+] 3 [)] → Memory A (5), namely memory E, is specified.

(3) Input to the memories

Numerical values and others are input to the memories in the following forms.

General form **[memory]** [=] **〈 expression 〉** [ENTER]

Example: [A] [=] 5 [*] 6 [ENTER] → Loading the answer of 5 * 6 (30) on the memory A

[Y] [=] [A] [+] [B] [ENTER] → Loading the contents of memory A and memory B on the memory Y.

[A] [(] 26 [)] [=] 3 [+] 9 [ENTER] → Loading the answer of 3 + 9 (12) on the memory A (26) (memory Z).

- When the memories are loaded with new data, they are cleared of their previous contents.

(4) Recalling the memory contents

The memory contents are recalled in the following form.

General form **[memory]** [ENTER]

Example: [A] [ENTER] → Recalling the contents of memory A

[A] [(] 18 [)] [ENTER] → Recalling the contents of memory A (18) (memory R)

8. Successive designation of expressions in manual calculation

In manual calculation, you can designate and solve two or more expressions in succession by punctuating them with comma. In this case, however, the computer displays the result of the final execution alone.

General form **〈 Expression 〉** [SHFT] [,] **〈 Expression 〉** [SHFT] [,] **〈 Expression 〉** [SHFT] [,] ..... [ENTER]

Example: When $A = \frac{5}{12-4}$, $B = \frac{87}{24}$, $C = \frac{12}{7+8}$, solve A*B/C=

Operation	Display	Note
[A] [=] 5 [/] [(] 12 [−] 4 [)]	A=5/(12−4)_	
[SHFT] [,] [B] [=] 87 [/] 24 [SHFT] [,]	A=5/(12−4),B=87/24,_	
[C] [=] 12 [/] [(] 7	/(12−4),B=87/24,C=12/(7_	
[+] 8 [)] [SHFT] [,]	−4),B=87/24,C=12/(7+8),_	
[A] [*] [B] [/] [C]	=87/24,C=12/(7+8),A*B/C_	
[ENTER]	2.83203125	

9. PLAYBACK function

You will get back in the display a portion of your original input to check and/or edit with this function. This function is activated by pressing the [►] or [◄] key right after the [ENTER] key in manual calculation.

Example 1: When an execution is finished without occurence of an error:

Operation	Display	Note
[A] [=] 19 [+] 54	A=19+54_	
[ENTER]	73.	
[►] or [◄]	A=19+54	Recall

The cursor shows up at the head of a display.
(The contents of the execution are displayed from the head in sequence.)

Example 2: When an error occurs:

Operation	Display	備考
[B] [=] [(] [(] 123456 [+]	B=((123456+_	
789012 [)] [*] 427 [/] 197	(123456+789012)*427/197_	
[)] [/] 0 [+] 139	+789012)*427/197)/0+139_	
[ENTER]	1.....	Occurrence of an error
[►] or [◄]	3456+789012)*427/197)/0+	Recall

The cursor appears where an error is detected.
(The contents of the execution is displayed to the extent that the display involves the information stored where the error has been detected.)

10. Editing expressions

Input exepressions, if recalled by the playback function before or immediately after their execution, can be arbitrarily edited (subjected to correction, insertion or deletion).
When you make a correction, insertion or deletion, follow the procedures given below.

Correction: Move the cursor to the position where information to be corrected is stored, by operating the [►] or [◄] key, and make a proper key operation anew.

Insertion: Move the cursor to the address position where you want to insert information, by the aid of the [►] or [◄] key, and press the [SHFT] and [INS] keys.

Then the contents of that address and the subsequent are individually shifted one step backward and an insertion mark (□) appears in the emptied address. The cursor still stays at that position, and you can enter an insert there.

Deletion: Move the cursor to the address position whose contents you want to delete, by pressing the [►] or [◄] key and press the [SHFT] and [DEL] keys. The contents in the said address are deleted and the subsequent contents are shifted one step forward.
The cursor still remains at the said position.

Example: When you make a mistake in inputting the expression below, edit as follows.

A=5+6∗ (21/SIN30)

Proper operation

[A] [=] 5 [+] 6 [∗] [(] 21 [/] [S] [I] [N] 30 [)] [ENTER]

(1) Correction when the [+] key is pressed by mistake instead of the [∗] key.

Operation	Display	Note
[A] [=] 5 [+] 6 [+]	A=5+6+_	The [+] key is indexed instead of the [∗] key.
[◄]	A=5+6+	The cursor goes down.
[∗]	A=5+6∗_	A proper key is indexed.
⋮		

(2) When you have forget to input the [S], [I] and [N] keys (Insertion of SIN)

Operation	Display	Note
DEG		
[A] [=] 5 [+] 6 [∗]	A=5+6∗_	
[(] 21 [/] 30 [)]	A=5+6∗ (21/30) _	SIN is not entered.
[◄] [◄] [◄]	A=5+6∗ (21/30)	The cursor goes down.
[SHFT] [INS]	A=5+6∗ (21/□30)	The steps where you want to enter SIN are emptied.
[SHFT] [INS] [SHFT] [INS]	A=5+6∗ (21/□□□30)	
[S] [I] [N]	A=5+6∗ (21/SIN30)	SIN is inserted.
[ENTER]	257.	

(3) When you have input 211 by mistake instead of 21 (Deletion of 1)

Operation	Display	Note
DEG		
[A] [=] 5 [+] 6 [∗] [(]	A=5+6∗ (_	
211 [/] [S] [I] [N] 30 [)]	A=5+6∗ (211/SIN30) _	
[◄] [◄] [◄] [◄]	A=5+6∗ (211/SIN30)	The cursor goes down to the position where you want to make an deletion.
[◄] [◄] [◄] [◄]	A=5+6∗ (211/SIN30)	
[SHFT] [DEL]	A=5+6∗ (21/SIN30)	1 is deleted.
[ENTER]	257.	

(4) When you have input 2 by mistake instead of 6 and executed the computation.

Operation	Display	Note
DEG [A] [=] 5 [+] 2 [*] [(]	A=5+2* (_	
21 [/] [S] [I] [N] 30 [)]	A=5+2* (21/SIN30) _	
[ENTER]	89.	
[◄]	A=5+2* (21/SIN 30)	The contents of execution are recalled.
[►] [►] [►] [►]	A=5+2* (21/SIN 30)	The cursor goes up.
6	A=5+6 * (21/SIN 30)	Correction
[ENTER]	257.	

(5) When you have input SIN 0 by mistake instead of SIN 30 and executed the computation (Occurrence of an error)

Operation	Display	Note
DEG [A] [=] 5 [+] 6 [*] [(]	A=5+6* (_	
21 [/] [S] [I] [N] 0 [)]	A=5+6* (21/SIN0) _	
[ENTER]	1	Error is detected.
[►]	A=5+6* (21/SIN 0)	The contents of execution are recalled.
[◄] [SHFT] [INS]	A=5+6* (21/SIN □0)	3 is inserted.
3	A=5+6* (21/SIN 30)	
[ENTER]	257.	

- **Cursor fast running function**
  If you keep the [►] or [◄] key pressed, so the cursor automatically goes up (rightward) or down (leftward) about one second later. The cursor moves 10 steps per one second. The cursor stops running fast as soon as the said keys are released. When editing long information, run the cursor fast to near the desired position and move it step by step by pressing the [►] or [◄] key.

〈 Number of input characters 〉

The computer once places data – numerals, alphabet, calculation instructions appearing on the display – that are entered through the keyboard, in the input buffer, decides, with the depression of the [ENTER] key, what instructions are entered, and executes input instructions after converting them to the corresponding codes.
The input buffer can contain a maximum of 80 characters. (1 character corresponds to a 1-digit display) when the buffer is loaded with 80 characters, the cursor flickers on the rightest place of the display. If further given an input in this state, the machine only provides a display of a character newly input in instead of a display of an existing 80th character, refusing to shift the display to the left.
A manual calculation is not properly executed, if its data contains over 80 characters including [ENTER] key. Therefore, you must limit its data scale to within 80 characters.
The contents recalled through the playback function refer to the information stored in the input buffer after converted to codes on the 1-instruction/1-step basis.

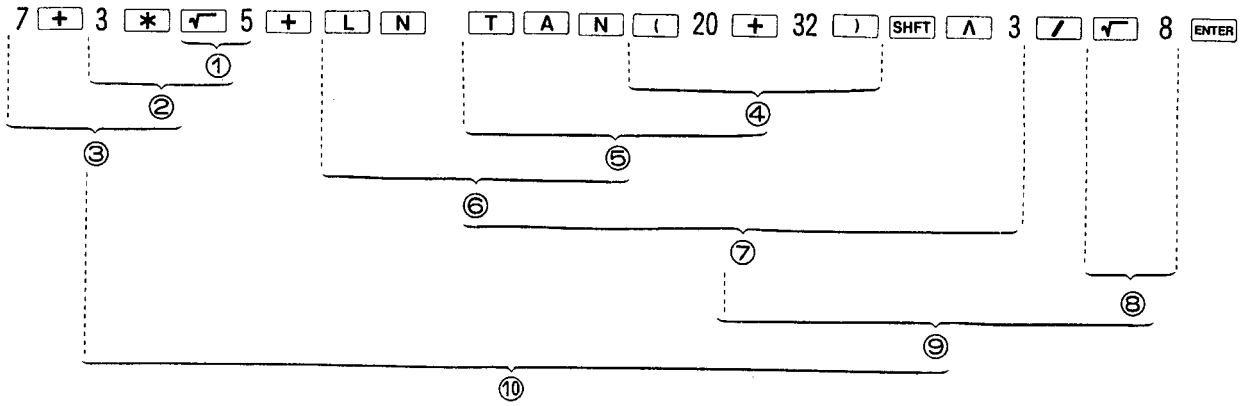
11. Priority of calculations

In principle, the computer performs calculations from the left to the right. However, those are individually given priority level depending on their kind; for example, function, multiplication or division has priority over addition or subtraction. The following shows the priority level of the computer.

1. Recalling π and fixed memories A through Z.
2. Recalling memories in the form of A ( ) (Recalling dimension memories)
3. Power directly preceded by a multiplication in which $*$ to be right before memory or π is omitted: 2A ∧ 3, for example.
4. Multiplication of which $*$ is omitted as 2A, πB or AB (see page 27.)
5. Functions (SIN, COS, TAN, ASN, ACS, ATN, LN, LOG, EXP, DMS, DEG, INT, ABS, SGN, $\sqrt{\ }$)
6. Power (/\ ) other than defined in 3 above.
7. Sign (+, −)
8. Multiplication and division ($*$, /)
9. Addition and division (+, −)
10. Logical computation (=, >, <, >=, <=, <>)

- Calculations in parenthesis will occur first. In multiple parenthesis, calculations in the innermost parentheses have priority over all the others.
- Compound functions (LN ABS A , EXP $\sqrt{\ }$ 8) are calculated from the right to the left.
- A string of powers, such as 3 ∧ 4 ∧ 2, is calculated from the right to the left.

Levels of pending operation



As seen from the example above, the computer performs computations following a given mathematical formula. But this presupposes that the computer has a place to temporarily store instructions or data (numerical values) that cannot be directly processed. Such a place is called a stack (a stack register). The PC-1211 has a 16-stage function stack and 8-stage data stack.

Example: Behavior of the both stacks experience during the execution of
1.2 + A * (3.5 + SIN B) ∧ A (25) ENTER
Where A = 2.4 , B = 30, A(25) = 3
Angular mode = DEG

Instruction	X register	Data stack				Function stack					
		1st stage	2nd stage	3rd stage	···	1st stage	2nd stage	3rd stage	4th stage	5th stage	···
1.2	1.2										
+	1.2	1.2				+					
A	2.4	1.2				+					
*	2.4	2.4	1.2			*	+				
(	2.4	2.4	1.2			(	*	+			
3.5	3.5	2.4	1.2			(	*	+			
+	3.5	3.5	2.4	1.2		+	(	*	+		
SIN	3.5	3.5	2.4	1.2		SIN	+	(	*	+	
B	30	3.5	2.4	1.2		SIN	+	(	*	+	
)	0.5	3.5	2.4	1.2		+	(	*	+		
	4	2.4	1.2			*	+				
^	4	4	2.4	1.2		^	*	+			
A(	4	4	2.4	1.2		A(	^	*	+		
25	25	4	2.4	1.2		A(	^	*	+		
)	3	4	2.4	1.2		^	*	+			
ENTER	64	2.4	1.2			*	+				
	153.6	1.2				+					
	154.8										

X register: Calculation register

As seen from the above table "A(" is placed in the function stack as one step. The computer permits to give a depth to parentheses up to 15 stages unless the capacity of the function stack is exceeded.

= Reference =

The computer permits you to input 2 * A, 3 * π or B * A (12), for example, in the form of 2A, 3π or BA(12) omitting the multiplication symbol * right ahead of memory or π. Such a form of multiplication has priority over functions, but when it is directly followed by a power, the power takes precedence over it.

Example: SIN 2A → equivalent to SIN (2 * A)

2πA$\wedge$3 → equivalent to 2 * π * (A$\wedge$3)

Expressions put in the machine as mentioned above – with multiplication instruction * omitted – are, however, executed also as containing the instruction. (*)

Example: Behavior of the stacks in the execution of 2ABC ENTER

If A = 3, B = 5, C = 7

Instruction	X register	Data stack			Function stack		
2	2						
A	3	2			*		
B	5	3	2		*	*	
C	7	5	3	2	*	*	*
ENTER	35	3	2		*	*	
	105	2			*		
	210						

PROGRAMMING

The programming language the PC-1211 uses is BASIC language. The BASIC language, a dialogue language for scientific computations, is said to be easiest to understand and use among a variety of programming languages and is widely used by people ranging from beginners to experts with programming.

1. What is a program calculation?

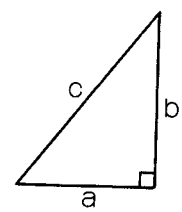
The manual calculation descrived on the preceding pages is entirely performed by manual operation, whereas the program calculation we will discuss hereafter is executed following the operational procedures (processing procedures) you have input in advance to the computer.
When computing for Pythagoras's theorem, for example, you must carry out the following operation.

Pythagoras's theorem

For a rectangular triangle, its three sides a, b and c have the following relations

$$c = \sqrt{a^2 + b^2}$$

where C is the opposite side to the right angle.



The manual calculation requires the following operation.
(when a = 3 and b = 4.)

Operation	Display	Note
Set to the RUN mode.		
[A] [=] 3 [ENTER]	3.	A is loaded with 3.
[B] [=] 4 [ENTER]	4.	B is loaded with 4.
[C] [=] [√] [(] [A] [*] [A]	C=√(A*A_	
[+] [B] [*] [B] [)]	C=√(A*A+B*B)_	
[ENTER]	5.	$\sqrt{A^2+B^2}$

(The above calculation can be of course accomplished by the operation:
[√] (3 * 3 + 4 * 4) [ENTER] . However, we use here the key operations as shown above for your easier understanding of the program given below.)

A series of these key operations can be programmed as follows.

[PROGRAM 1]

Programming	Note
10 : INPUT A , B	Input instruction
20 : C=√(A*A+B*B)	Operation instruction
30 : PRINT C	Output instruction
40 : END	End instruction

Basically, the program is completed if you add input instruction (INPUT), output instruction (PRINT) and end instruction (END) to the step of operational processing in the manual calculation procedures.

Input instruction: Commands to input data to the memory.
Output instruction: Commands to display calculation results and others.

The tables given below reveals how to write the program in the PC-1211 and execute it.

Operation	Display	Note
[Writing]	┌ Prompt symbol	
Set to the PRO mode.	>	Symbol "PRO" appears with the press of the MODE key.
N E W ENTER	>	Program memory is cleared.
10 I N P U T	10 I N P U T _	10th line is put in. (Input instruction)
A SHFT , B	10 I N P U T A , B _	
ENTER	10 : I N P U T A , B	10th line is written in.
20 C = √ (A	20 C = √ (A _	20th line is put in (Operation instruction)
* A + B	20 C = √ (A * A + B _	
* B)	20 C = √ (A * A + B * B) _	
ENTER	20 : C = √ (A * A + B * B)	20th line is written in.
30 P R I N T	30 P R I N T _	30th line is put in. (Output instruction)
C	30 P R I N T C _	
ENTER	30 : P R I N T C	30th line is written in.
40 E N D	40 E N D _	40th line is put in (End instruction)
ENTER	40 : E N D	40th line is written in.
[Execution]		
Set to the RUN mode.		
MODE MODE MODE	>	The RUN mode is chosen.
R U N	R U N _	Execution start instruction (RUN) is put in.
ENTER	?	Execution is started; the display tells you to input a variable.
3	3 _	A variable 3 is put in.
ENTER	?	The variable is written in (3 is loaded on memory A); the display tells you to input a following variable.
4	4 _	A variable 4 is put in.
ENTER	5.	The variable is written in (4 is loaded on memory B); calculation result is displayed.
ENTER	>	Execution is terminated.

Repeat these operations inputting different values. So you can calculate according to Pythagoras's theorem any number of times.

Thus, once a program is written, you can execute it simply the number of times you want.

2. Wirting programs

When you write programs through the keyboard, put the computer in the PRO mode and follow the same procedures as in manual calculation. Consult the procedures for writing "PROGRAM 1".

(1) Preparation

When writing a new program, you are requested to clear the program memory by the aid of a NEW command.

However, this is not true of the case where you write in a program in succession with the preceding one.

[Procedures] (1) Designate the PRO mode.

(2) [N] [E] [W] [ENTER]

All contents of the program and data memories will be cleared with above (2) operation.

(2) Writing in

Detailed below is the writing of the said PROGRAM 1.

Step	Operation	Display	Note
	Setting to the PRO mode		
1	[N] [E] [W] [ENTER]	>	The program memory is cleared.
	A colon (:) does not need to be put in.		
2	10 [I] [N] [P] [U] [T]	10 INPUT_	10th line is put in. (The input is once placed in the input buffer.)
3	[A] [SHFT] [,] [B]	10 INPUTA , B_	
4	[ENTER]	10 : INPUT A , B A display of colon, Space	The above input is written in the program memory at the push on the [ENTER] key.
5	20 [C] [=] [√] [(] [A]	20C=√ (A_	20th line is put in. (The input is once placed in the input buffer.)
6	[*] [A] [+] [B]	20C=√ (A*A+B_	
7	[*] [B] [)]	20C=√ (A*A+B*B) _	
8	[ENTER]	20 : C=√ (A*A+B*B)	The program memory is loaded with the said input.
9	30 [P] [R] [I] [N] [T]	30PRINT_	30th line is put in. (The input is once placed in the input buffer.)
10	[C]	30PRINTC_	
11	[ENTER]	30 : PRINT C	The program memory is loaded with that input.
12	40 [E] [N] [D]	40END_	40th line is put in. (The input once enters the input buffer.)
13	[ENTER]	40 : END	The program memory is loaded with the input.

As seen from the above table, the line contents are, when put in the computer, once placed in the input buffer and loaded on the program memory at the push of the [ENTER] key.

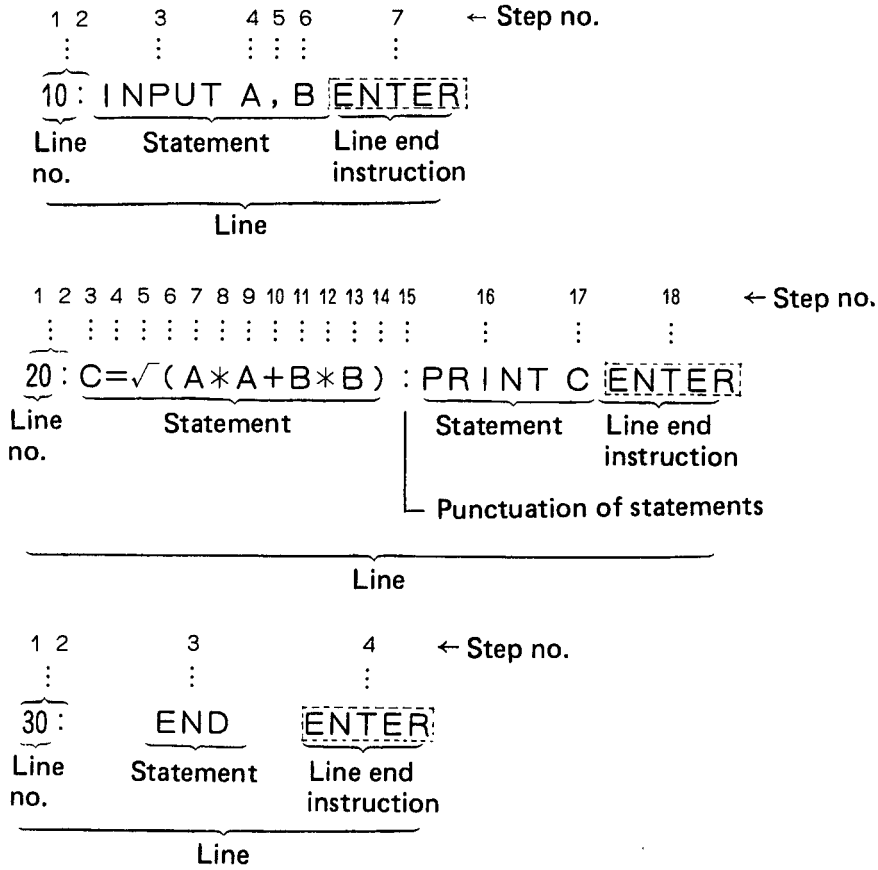
At this time imperative statements such as INPUT, PRINT are converted into corresponding instruction codes of 1-step capacity before placed in the memory (one instruction consists of 1 step.) Information thus written in the program memory appears on the display; a colon (:) appearing behind the line number together with that information shows the coincidence of the displayed information and the contents of program memory. The colon itself is not present in that memory. Besides, each of imperative statements such as INPUT and PRINT is followed by a space.

When the computer cannot display the whole line in 24 digits because it is long, it displays its contents — including the line number — from the beginning to the last instruction to be completely displayed.

= **Reference** =

The computer needs to be supplied with programs expressed in rows; every row is called a line, composed of a line number, label and statements.

Example: Composition of a program


1 2 3 4 5 6 7 ← Step no.
10 : INPUT A , B ENTER
Line no.
Statement
Line end instruction
Line
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ← Step no.
20 : C=√(A*A+B*B) : PRINT C ENTER
Line no.
Statement
Statement
Line end instruction
Punctuation of statements
Line
1 2 3 4 ← Step no.
30 : END ENTER
Line no.
Statement
Line end instruction
Line

[Line]

- Lines must be headed without exception by integral line numbers ranging from 1 to 999.
- Ending of every line occurs by inputting the ENTER key. The ENTER instruction is represented by a space in terms of display. (Nothing appears.)

[Statement]

- One line consists of one or more statements (statement instructions on the BASIC language basis).
- Statements are divided by colons (:).

[Step]

One statement consists of one or more operation instructions, each of those instructions having a capacity of one step.

Instructions such as LN, SIN and INPUT instructions are processed as one-step information to be written in the program memory, though represented in terms of display by two to six characters.

[Label]

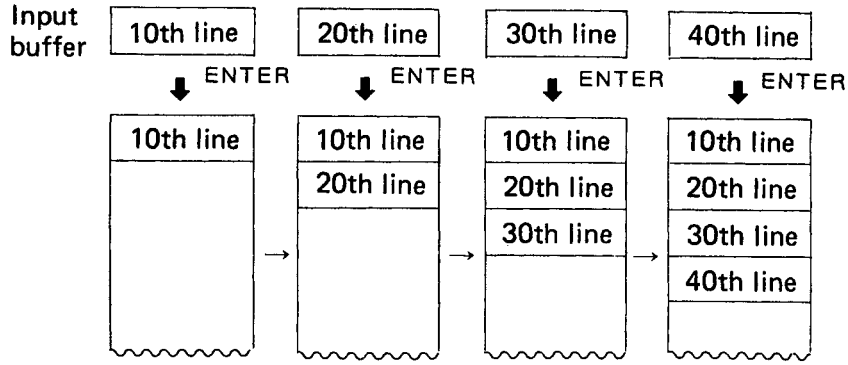
Characters (letters, numerals, symbols) are written as placed between two quotation marks following a line number. The label serves as sign of program jump, etc.

Note: Each of line numbers 1 through 999 is written without exception as two-step information in the program memory.

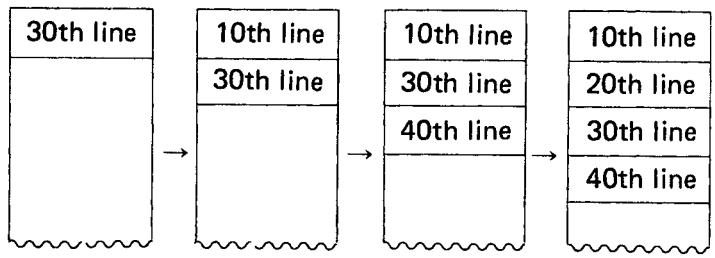
Although not present in a program memory, a colon (:) that follows every line number is automatically displayed immediately after programs have been written or when these are recalled.

When loaded with programs, the program memory exhibits the following change in its contents.

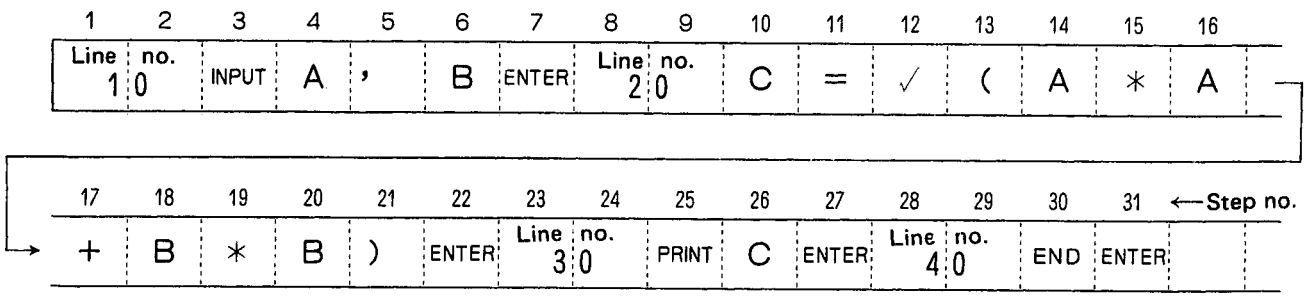
An input is once placed, on the 1 character = 1 step basis, in the input buffer and is written, when the [ENTER] key is pushed, in the program memory after converted into the form on 1 instruction = 1 step basis.



- One line of a program can comprise a maximum of 80 steps.
  Lines are, even if they are not written in numerical order — 30th line → 10th line → 40th line → 20th line, for example, stored in numerical order into the program memory.



However, the real program memory has not a separate storage area for every line unlike shown above, but stores programs step by step in a row, having a capacity of 1424 steps.


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Line no. 1 0 INPUT A , B ENTER Line no. 2 0 C = √ (A * A
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 ←Step no.
+ B * B) ENTER Line no. 3 0 PRINT C ENTER Line no. 4 0 END ENTER

3. Checking stored programs

You must check that programs are properly stored.
The computer brings programs on the display, whenever these are put in it through its keyboard, permitting you to check every input.
After the completion of program writing, you can check written programs following the procedures described below.

[Procedures]

① Select the PRO mode.
② Recall the line you want to check, by pressing the [↓] or [↑] key.
Or recall the intended line with a LIST command. (Consult the paragraph "LIST COMMAND" on page 73.)

③ Check instructions on the display.
The computer provides, if the [►] key is used as follows, a display of within 24-character information headed by the start of a line, when its display cannot show the contents of that line at one time: Operate the [►] key to move the cursor to the right. When the cursor reaches the right end, operate the [►] key in succession. Then the whole display is shifted step by step leftward being followed by the subsequent information. (When a line is long, you may run the display fast by keeping the [►] key depressed.)

Example: Checking programs

Operation	Display	Note
Setting to the PRO mode	>	
[↓]	10 : INPUT A , B	Lines are recalled in numerical order.
[↓]	20 : C= (A*A+B*B)	
[↓]	30 : PRINT C	
[↓]	40 : END	
[↑]	30 : PRINT C	Lines on the display are brought back to the memory.
[↑]	20 : C= (A*A+B*B)	
[↑]	10 : INPUT A , B	

- If you maintain the [↓] or [↑] key depressed for about one second, the display automatically shows the next line or the preceding line. (Line fast running function).

Note: When the program memory is loaded with nothing, pressing the [↓] or [↑] key or the execution of a LIST command brightens the prompt symbol (>).

4. Program correction

When you find errors in stored programs, follow the below-mentioned procedures for correction.

(1) Partial correction

[Procedures]

① Select the PRO mode.
② Bring the line you want to correct on the display by the aid of the [↓] or [↑] key, or a LIST command.
③ Move the cursor to the step you want, by pressing the [►] or [◄] key.
④ Make a correction, insertion or deletion in the same way as shown in the paragraph "Editing expressions" on page 23.
⑤ When the correction is finished, press the [ENTER] key. This will bring the corrected program back to the program memory.

Example: A program identical to PROGRAM 1 is set up by correcting the following program.

```
10 : INPUT A , B
20 : C=√(A*A+B+B)
30 : PAUSE C
40 : END
```

⟶

```
20 : C=√(A*A+B*B)
30 : PRINT C
```

Step	Operation	Display	Note
	Setting to the PRO mode	)	
1	[↓] [↓]	20 : C=√ (A∗A+B+B)	A line you want to modify is recalled.
2	[►]	20 C=√ (A∗A+B+B)	The cursol appears on the display.
3	[►] ..... [►]	20 C=√ (A∗A+B+B)	The cursor moves to the position where a correction is made.
4	[∗]	20 C=√ (A∗A+B∗B)	Correction
5	[ENTER]	20 : C=√ (A∗A+B∗B)	Remember to press [ENTER] .
6	[↓]	30 : PAUSE C	A line you want to modify is recalled.
7	[◄]	30 PAUSE C	The cursor appears on the display.
8	[P]	30 PC	} Correction
9	[R]	30 PR_	
10	[I] [N] [T] [C]	30 PRINTC_	
11	[ENTER]	30 : PRINT C	Writing

- Pressing once the [►] or [◄] key after the recall of a line recalls the cursor. (Steps 2 and 7 in the above table)
  The cursor shows up at the head of first statement; a colon (:) behind line number disappears to make a space there.

- Reaching a 1-step imperative statement such as INPUT or PRINT, the cursor appears only at its first character.
  And if something is put there, the whole characters of that statement disappears from the display. (Step 8 above)

(2) Inserting lines

When inserting lines into written programs, follows the procedures below.

[Procedures]

(1) Select the PRO mode.
(2) Input a line. This line must be given a line number equal to the numeral that stays between the line numbers of those lines which precedes and follows the desired place.
When you wish to insert a new line between 10th and 20th lines, you must give it a line number ranging from 11 to 19.
(3) Press the [ENTER] key. That new line is then written in the program memory.

Example: Insertion of PAUSE A, B between the 10th and 20th lines of PROGRAM 1. (Line number is 15.)

```
10 : INPUT A , B
20 : C=√(A∗A+B∗B)
30 : PRINT C
40 : END
```

15 : PAUSE A , B

Operation	Display	Note
Setting to the PRO mode.		
15 [P] [A] [U] [S] [E]	15PAUSE_	15th line is put in.
[A] [SHFT] [,] [B]	15PAUSEA,B_	
[ENTER]	15 : PAUSE A , B	Writing (insertion) of 15th line
[CL] [↓]	10 : INPUT A , B	Checking
[↓]	15 : PAUSE A , B	
[↓]	20 : C=√(A*A+B*B)	

(3) Deleting lines

To delete a certain line from the corresponding stored program, follows the below-mentioned procedures.

[Procedures]

(1) Select the PRO mode.

(2) Input only the line number of a line you want to delete, and press the [ENTER] key.

Example: Deletion of the 15th line is deleted.

Operation	Display	Note
Setting to the PRO mode.		
15	15_	Inputting the line number of a line to be deleted.
[ENTER]	>	15th line is deleted.
[↓]	10 : INPUT A , B	Checking
[↓]	20 : C=√(A*A+B*B)	

5. Executing programs

You must execute programs in the RUN or DEF mode.

[Procedures]

① Select the RUN mode.

② Press the [R] , [U] , [N] and [ENTER] keys. The computer starts the execution of programs.

③ When program execution stops with INPUT instruction (The "?" symbol then appears.), input data and push the [ENTER] key.

④ When program execution stops with PRINT instruction (Calculation result is displayed), push the [ENTER] key without inputting data.

⑤ Program execution comes to an end as soon as END instruction is read out. (The prompt symbol then appears.)

Example: Execution of program 1

(1) When A = 12.3, B = 15.7

(2) When A = 36, B = 27

Operation	Display	Note
Setting to the RUN mode.		
[R] [U] [N]	RUN_	Inputting RUN instruction.
[ENTER]	?	Execution starts.
Refer to above ③ { 12.3	12. 3_	Inputting data (A)
[ENTER]	?	
③ { 15.7	15. 7_	Inputting data (B)
[ENTER]	19. 94442278	Result appears.
Refer to above ④ { [ENTER]	>	Execution ends.
[R] [U] [N] [ENTER]	?	Execution starts.
36 [ENTER]	?	
27 [ENTER]	45.	Result appears.
[ENTER]	>	Execution ends.

6. Debugging programs

The debugging serves the checking to see if prepared programs are working properly. Programs are successively executed by one line to check the progress.

Example: Debugging of PROGRAM 1
When A = 36, B = 27

Operation	Display	Note
Setting to the RUN mode		
D E B U G ENTER	?	DEBUG command commences debugging.
36	36_	Inputting of data
↧	?	INPUT instruction for 10th line is executed.
27	27_	After the execution of the above instruction, the machine displays the line number and stops.
↧	10 :	
↥ (kept pressed)	10 : INPUT A , B_	Checking of executed instruction; the cursor indicates the executed instruction — ENTER in this example. (i.e. space)
Release the ↥	>	The prompt symbol appears after the display of 10th line.
↧	20 :	After the debugging of 20th line, its line number is displayed.
↧	45.	Display of calculation result (execution of PRINT statement)
A ENTER	36.	Checking of memory contents
B ENTER	27.	
↧	30 :	The computer stops after debugging 30th line.
↧	>	Debugging ends.

(For DEBUG command, consult page 72.)

In the debugging, as shown above, the ↧ key serves the execution of instructions on each line.

While the execution is stopping, you can check that the memory is properly loaded with data, through the manual opertion, such as A ENTER or B ENTER .

The ↧ key activates, even if pushed after the above manual operation, the debugging operation. Again, when the execution stops, the cursor indicates the step where the computer is resting and the corresponding program appears on the display while the ↥ key is being depressed.

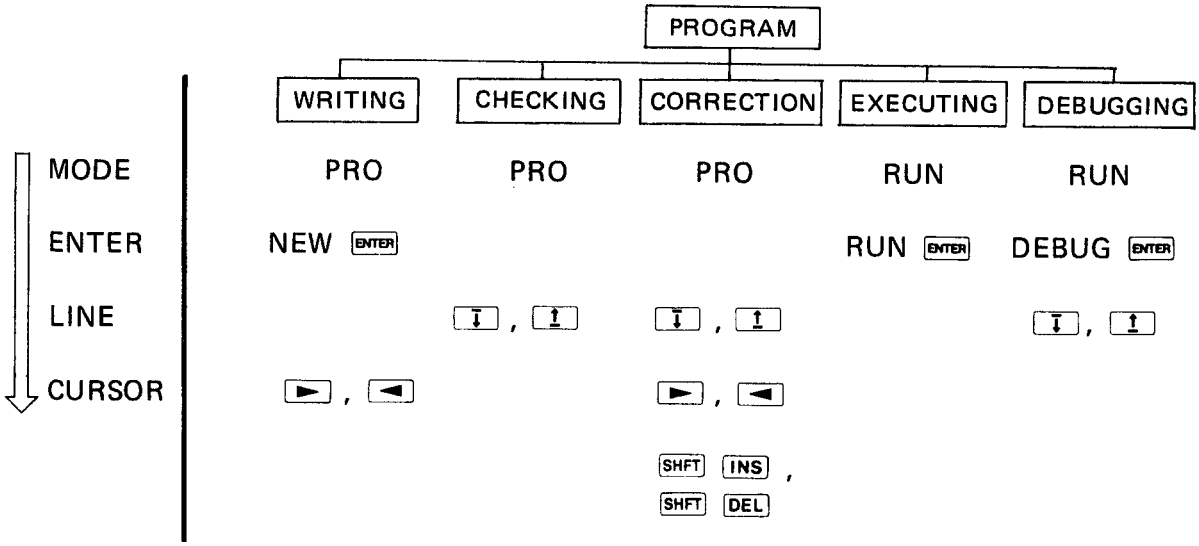
- To interrupt the debugging and resume the usual operation, let the machine execute CONT command.

  C O N T ENTER

[Rapid debugging]

If the ↧ key remains pressed for about one second, the machine will stop debugging and starts to execute programs in its usual state.

If the ↧ key is released, the machine will resume the debugging as soon as it passes the line executed then.



7. Defined program

When more than two programs are written in the program memory, others than its first one are usually executed by the key operation: R U N [line number] ENTER . If its keys including A 、 S and D are defined by assigning those programs to them, the computer begins, only by operating keys as SHFT A or SHFT S in the DEF mode, to execute the corresponding assigned program.

To assign programs to certain keys, you must write the labels of those keys at the heads of the programs you want to assign: "A" in the case of the key A , for example. (Place the labels right after the entry of line number without fail.) This will cause the PC-1211 to begin the program execution, only by making the key operation, such as SHFT A , SHFT S , in the DEF mode, from the line headed by the label of a key then pushed.

- The following 18 keys are definable.

A 、 S 、 D 、 F 、 G 、 H 、 J 、 K 、 L 、 = 、
Z 、 X 、 C 、 V 、 B 、 N 、 M 、 SPC

Example: The following defined programs are written in and executed.

PROGRAM 2

Programming	Note
10 : "A" : INPUT A , B 20 : C=√(A∗A+B∗B) 30 : PRINT C 40 : END	Label A $C=\sqrt{a^2+b^2}$: Pythagoras's theorem
50 : "S" : INPUT D 60 : E=4/3∗π∗D^3 70 : PRINT E 80 : END	Label S $V=\frac{4}{3}\pi r^3$: Volume of sphere
90 : " " : INPUT F , G , H 100 : I=√(F∗F+G∗G−2∗F∗G∗COS H) 110 : PRINT I 120 : END	Label SPC (space) $C=\sqrt{a^2+b^2-2ab\cos\theta}$: Law of cosine

[Writing]

Operation	Note
Setting to the PRO mode [N] [E] [W] [ENTER]	
A colon behind a label can be eliminated. 10 [SHFT] ["] [A] [SHFT] ["] [SHFT] [:] [I] [N] [P] [U] [T] [A] [SHFT] [,] [B] [ENTER] 20 [C] [=] [√] [(] [A] [∗] [A] [+] [B] [∗] [B] [)] [ENTER] 30 [P] [R] [I] [N] [T] [C] [ENTER] 40 [E] [N] [D] [ENTER]	Label A
50 [SHFT] ["] [S] [SHFT] ["] [SHFT] [:] [I] [N] [P] [U] [T] [D] [ENTER] 60 [E] [=] [4] [/] [3] [∗] [SHFT] [π] [∗] [D] [SHFT] [^] [3] [ENTER] 70 [P] [R] [I] [N] [T] [E] [ENTER] 80 [E] [N] [D] [ENTER]	Label B
90 [SHFT] ["] [SPC] [SHFT] ["] [SHFT] [:] [I] [N] [P] [U] [T] [F] [SHFT] [,] [G] [SHFT] [,] [H] [ENTER] 100 [I] [=] [√] [(] [F] [∗] [F] [+] [G] [∗] [G] [−] [2] [∗] [F] [∗] [G] [∗] [C] [O] [S] [H] [)] [ENTER] 110 [P] [R] [I] [N] [T] [I] [ENTER] 120 [E] [N] [D] [ENTER]	Label SPC (Space)

[Execution]

Operation	Display	Note
Remember to select the DEF mode		
SHFT A	?	Execution of a program labeled A starts.
(When A = 4) 4 ENTER	?	
(When B = 3) 3 ENTER	5.	Result
ENTER	>	End
SHFT S	?	Execution of a program labeled S starts.
(When D = 2) 2 ENTER	33. 51032164	Result
ENTER	>	End
D E G R E E ENTER	>	Angular mode is set to degree.
SHFT SPC	?	Execution of a program labeled SPC starts.
(When F = 12) 12 ENTER	?	
(When G = 14) 14 ENTER	?	
(When H = 30) 30 ENTER	7. 001104508	Result
ENTER	>	End

To resume the execution that is interrupted with INPUT or PRINT instruction, push the ENTER key as the above example shows.

- When the identical label are given to two or more lines, the line smallest in line number is executed with priority.
  In the case 10th and 40th lines are headed by label A, for example, programs on the 10th line are executed with the operation: SHFT A .
- Inputting an undefined key causes an error. (Error code: 2)

RESERVABLE KEY

In the descriptions given so far, characters are all put one by one in the machine by key operation both in manual calculation and programming. The computer also enables you to assign key operations frequently used to keys such as [A] , [S] and [D] for reserve. This function permits you to recall, in manual calculation or programming, reserved key operations through a simple operation.

When the operation [P] [R] [I] [N] [T] is assigned to the [A] key for reserve, for example, the operation [SHFT] [A] forces the machine to give a display of the said operation PRINT.

The following keys are available for reserve:

([A]、 [S]、 [D]、 [F]、 [G]、 [H]、 [J]、 [K]、 [L]、 [=]、 [Z]、 [X]、
[C]、 [V]、 [B]、 [N]、 [M]、 [SPC]

1. Reserve for reservable keys

Reserve for reservable keys are made in the 48-step reserve memory, which allows reserve on above 18 keys (Reserve program).

To write reserve programs, select the RESERVE mode and follows the same procedures as in manual calculation.

(1) Preparation

Writing reserve program anew requires to clear the reserve memory using a NEW command. This is, however, not true of writing reserve programs following the preceding one already written.

[Precedures]

(1) Select the RESERVE mode.

(2) [N] [E] [W] [ENTER]

The operation clears the whole reserve memory.

(2) Writing

Example: Reserve the following key operation:

"COS" to [A]

"A * A + B * B" to [S]

"RUN 130" to [Z]

Operation	Display	Note
Setting to the RESERVE mode		
[N] [E] [W] [ENTER]	> Colon is displayed automatically.	Reserve memory is cleared.
[SHFT] [A]	A : _	Key is specified.
[C] [O] [S]	A : COS_	Input
[ENTER]	A : COS	Writing (the cursor disappears.)
[SHFT] [S]	S : _	Key is specified.
[A] [*] [A] [+] [B] [*] [B]	S : A*A+B*B_	Input
[ENTER]	S : A*A+B*B	Writing
[SHFT] [Z]	Z : _	Key is specified
[R] [U] [N] [1] [3] [0]	Z : RUN130_	Input
[ENTER]	Z : RUN 130	Writing

- The push of the [ENTER] key causes an error, when the reserve memory (48 steps) cannot receive the input at all. (Error code: 4)

2. Use of reservable keys

The reservable keys are used in the PRO or RUN mode.

Example 1: Manual calculation by using the reservable keys to which reserve programs have been alloted in (2) of 1 above.

Operation	Display	Note
Setting to the RUN mode. Setting the angular mode to DEG		Degree is specified.
[SHFT] [A]	COS _	
60	COS 60_	
[ENTER]	0. 5	
[A] [=] 32 [SHFT] [,]	A=32 , _	
[B] [=] 53 [ENTER]	53.	
[SHFT] [S]	A*A+B*B_	
[ENTER]	3833.	$32^2+53^2=$

Example 2: Writing and execution by using the reservable keys to which reserve programs have been assigned in (2) of 1 above.

Operation	Display	Note
[Writing]		Low of cosine
Setting to the PRO mode		$C=\sqrt{a^2+b^2-2ab\cos\theta}$
130 [I] [N] [P] [U] [T]	130 INPUT_	
[A] [SHFT] [,] [B]	130 INPUTA , B_	
[SHFT] [,] [D] [ENTER]	130 : INPUT A , B , D	
140 [C] [=] [√] [(]	140C=√ (_	
[SHFT] [S]	140C=√ (A∗A+B∗B_	Recall of reserve
[−] [2] [∗] [A] [∗]	140C=√ (A∗A+B∗B−2∗A∗_	
[B] [∗] [SHFT] [A]	0C=√(A∗A+B∗B−2∗A∗B∗COS _	Recall of reserve
[D] [)] [ENTER]	140 : C=√ (A∗A+B∗B−2∗A∗B∗	
150 [P] [R] [I] [N] [T]	150PRINT_	
[C] [ENTER]	150 : PRINT C	
160 [E] [N] [D] [ENTER]	160 : END	
[Execution]		
Setting to the RUN mode		
Setting the angular mode to DEG		
[SHFT] [Z]	RUN130_	Program execution starts.
[ENTER]	?	
(When A = 10) 10 [ENTER]	?	
(When B = 12) 12 [ENTER]	?	
(When D = 60) 60 [ENTER]	11. 13552873	

Example 3: When inserting reservable keys in the course of manual calculation or programming.

Operation	Display	Note
Setting to the RUN mode		
5 [∗] 6 [∗] 60	5∗6∗60_	
[◄] [◄] [SHFT] [INS]	5∗6∗□60	Spacing by one step
[SHFT] [A]	5∗6∗COS 60	COS is inserted as 1-step instruction.
[◄] [SHFT] [INS] [SHFT] [INS]	5∗6∗□□COS 60	Spacing by two steps
[SHFT] [S]	5∗6∗A∗A+B∗BCOS 60	When the number of steps for reservable key to be inserted is larger than that of insertion codes, the necessary number of steps are secured to insert the remaining code.
[∗]	5∗6∗A∗A+B∗B∗60	
[SHFT] [INS]		
[SHFT] [INS] [SHFT] [INS]	5∗6∗A∗A+B∗B∗□□□60	Spacing by three steps
[SHFT] [A]	5∗6∗A∗A+B∗B∗COS □□60	When the number of steps for reservable key to be inserted is smaller than that of insertion codes, the empty insertion codes remain filled with nothing.

3. Checking reserve programs

To check what information is assigned to reservable keys, specify those keys in the RESERVE mode.

Example: Checking reserve programs written in (2) of 1 above.

Operation	Display	Note
Setting to the RESERVE mode		
[SHFT] [A]	A : COS	Display of reserve.
[SHFT] [S]	S : A∗A+B∗B	Display of reserve.
[SHFT] [D]	D : _	When nothing is reserved
[SHFT] [Z]	Z : RUN 130	Display of reserve

4. Correction of reserve programs

Described below is the method of correcting written reserve programs or of reserving different key operations.

Example: "A : COS", "S : A∗A + B∗B" and "Z : RUN 130" reserved above are reserved anew after modified as "A : SIN", S : LOG A" and "Z : RUN 50."

Operation	Display	Note
Setting to the RESERVE mode		
[SHFT] [A]	A : COS	Recall of reserve
[◄]	A : COS	Recall of cursor
[S] [I] [N]	A : S I N_	Inputting through keys
[ENTER]	A : S I N	Writing
[SHFT] [S]	S : A∗A+B∗B	Recall of reserve
[►]	S : A∗A+B∗B	Recall of cursor
[L] [O] [G] [A]	S : LOGAB∗B	Inputting through keys
[SPC] [SPC] [SPC]	S : LOGA _	A necessary instructions are cleared by spacing.
[ENTER]	S : LOG A	None of spaces are written in the reserve memory.
[SHFT] [Z]	Z : RUN 130	Recall or reserve
[►] [►]	Z : RUN 130	The cursor moves
[5] [0] [SPC]	Z : RUN 50 _	Inputting
[ENTER]	Z : RUN 50	Writing

5. Deleting reserve programs

The table below reveals how to delete the reserve for a certain key alone.
(cf. The operation of [N] [E] [W] [ENTER] key clears all reserve memories.)
Example: "RUN 50" reserved for [Z] is deleted.

Operation	Display	Note
Setting to the RESERVE mode		
[SHFT] [Z]	Z : RUN 50	Recall of reserve
[►]	Z : RUN 50	Recall of cursor
[SPC]	Z : 50	All instructions are replaced with spaces.
[SPC] [SPC]	Z : _	
[ENTER]	>	Reserve is deleted and prompt symbol appears.

6. Configuration of reserve programs

Example: Operation [S] [I] [N] [A] [+] [C] [O] [S] [B] is reserved for [A] key.

```
1    2    3 4   5    6   ← Step no. of reserve memory
:    :    : :   :    :
A : SIN  A+COS  B
↑
└ Reservable key label
```

Operation [I] [N] [P] [U] [T] is reserved for [B] key.

```
1    2    ← Step no.
:    :
B : INPUT
↑
└ Reservable key label
```

In the reserve memory are also placed labels of keys, such as [A] and [B], each of which needs 1-step capacity. (Colons behind key labels are not put in the reserve memory, however.)

VARIABLES

1. What is a variable?

While the mathematics defines a variable as: number or volume that can vary taking various values in certain relations or within a certain range, a memory (data memory) in program is termed as a "variable".

In this computer variables are divided into fixed memories (26 pieces) and flexible memory as described in the paragraph 7. "Calaculations using memories" of chapter "Manual Calculation" (See page 15) The data memories stores not only numerical values, but they also stores items composed of characters such as person's name or item name.

(1) Numerical variable

While or when stored with numerical values, a data memory is called a numerical variable, being specified by labeling it as A, B, C, A(1) or A(28).

(2) Character variable

While or when stored with sequences of characters, including letters, blanks, numbers, special symbols, a data memory is called a character variable, being specified by labeling it as A$; B$; C$ or A$(1), all of which are accompanied by symbol $. (One data memory can contain a maximum of seven characters.)

```
10 : INPUT  A$, B$
20 : PRINT  A$, B$
  :
```

Operation	Display	Note
Setting to PRO mode		
[N] [E] [W] [ENTER]	>	Clears the memory
10 [I] [N] [P] [U] [T]	10 INPUT_	
[A] [SHFT] [$] [SHFT] [,]	10 INPUT A$, _	
[B] [SHFT] [$]	10 INPUT A$, B$_	
[ENTER]	10 : INPUT A$, B$	
20 [P] [R] [I] [N] [T]	20 PRINT_	
[A] [SHFT] [$] [SHFT] [,]	20 PRINT A$,_	
[B] [SHFT] [$]	20 PRINT A$, B$_	
[ENTER]	20: PRINT A$, B$	
Change the PRO mode to RUN		
[R] [U] [N] [ENTER]	?	
[S] [M] [I] [T] [H]	SMITH_	
[ENTER]	?	A$ is loaded with "SMITH"
26438	26438	
[ENTER]	SMITH 26438	B$ is loaded with "26438."

Note) If the variable loaded with numerical value is specified as a character variable or if a character variable or if a character variable is specified to store a numerical value, an error (error code: 1) occurs. This is true of the reverse cases. The above-mentioned practice causes no errors, when variables are cleared: that is, when numerical variable is loaded witho 0 or character variable is loaded with no character.

2. Specifying variables

(1) Fixed memory

1. Specify fixed memories simply by pressing a single key as [A] , or two or more keys as [B] [SHFT] [$] .

   Example: [A] → Numerical variable A is specified.
   [B] [SHFT] [$] → Character variable B$ is specified.

2. Fixed memories A through Z or A$ through Z$ are individually given serial numbers 1 through 26, being able to be specified by inputting codes such as A(1) and A(5) or A$(1) and A$(5).

   Example: A (1) → Numerical variable A is specified.
   A$ (1) → Character variable A$ is specified.
   A (48−25) → Numerical variable A(23), namely W, is specified.
   A$ (3 ∗4) → Character variable A$(12), namely L$, is specified.

- When the above specification is made in the form of A ( ) or A$ ( ), only the integer part of parenthesized value is effective.
- Memories specified in the form of A ( ) or A$ ( ) are especially called dimension memories.
- In designation of memories, A through Z and A(1) through A(26) specify the corresponding identical memories respectively; and A$ through Z$ and A$(1) through A$(26) also specify after all the corresponding same memories as above.
  When B or A(2) is entered, for example, data memory B is specified as numerical variable; when B$ or A$(2) is entered, the same data memory as above, namely memory B, is specified as character variable.

(2) Flexible memory

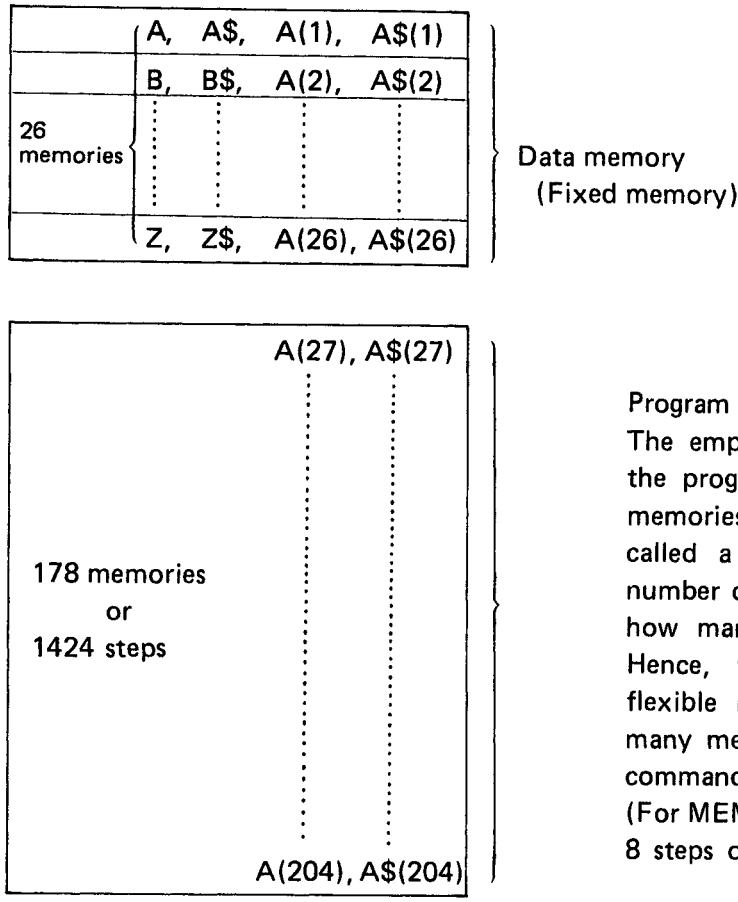
Flexible memory is specified in the form A ( ) or A$ ( ) in the same manner as in 2 of (1). But when the value in parentheses is smaller than 27, this memory is not specified, since number of fixed memories is 26.

Example: A(27) → Specification of numerical variable A(27) (flexible memory)
A$(19 x 2) → Specification of character variable A$(38) (flexible memory)

Note)

Memory
- Data memory (Variable)
  - Numerical memory (Numerical variable)
  - Character memory (Character variable)
- Program memory

The memories of the PC-1211



Program memory
The empty area in the program memories that the programs are stored are also used as data memories. The Data memory in this case is called a flexible memory. Accordingly, the number of alloted memories varies depending on how many steps are used to store programs. Hence, you are requested, when using the flexible memory, to ascertain in advance how many memories can be alloted by using a MEM command.
(For MEM command, refer to page 74.)
8 steps of program equals to one data memory.

(3) Indirect designation

The indirect designation of memories (variables) is the method of designating an arbitrary memory (numerical variable) corresponding to its contents.
The indirect designation is made in the form of A(B) or A$(B); that is, a numerical variable is put in a set of parentheses.
This method enables to specify all data memories (variables) according to their contents – only the integer parts of them are effective.

Example: A(A) → A numerical variable that is given a serial number corresponding to variable A is specified.

A$(A(3)) → A character variable that is given a serial number corresponding to variable A(3); namely C, is specified.

The indirect designation displays its advantageous effects in the following case.

Example: Programs whose data is put in variables B through Z.

```
10 : INPUT  B, C, D, ·····, Z
  ⋮
```

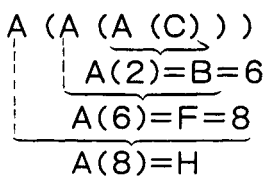
When variables B through Z are directly specified in 10th line.

```
10 : FOR  A=2TO  26
20 : INPUT  A (A)
30 : NEXT  A
  ⋮
```

The value of A varies from 2 to 26; programs stored in the variables corresponding those numbers are repeatedly executed.
In response to the execution are specified variables B through Z in sequence.

The indirect method can provide a depth of a maximum 15 stages by specifying dimension memories in sets of parentheses.

Example: When C = 2, B = 6, F = 8,
A (A(A(C))) → Variable H is specified.



Note) In designation of dimension memories, when a specified value is below 1 or exceeds the area within which the flexible memory is specified, an error occurs. (Error code: 4)

3. Inputting to variables

The memories (variables) are loaded with numerical values or characters in the following forms.

General form (1) **[Numerical variable] = ⟨ Expression ⟩**
The value of ⟨ expression ⟩ is put in a numerical variable specified on the left side of the above equation.
Note: ⟨ expression ⟩ covers also a numerical variable.

General form (2) **[Character variable] = "Characters"**
Characters (letters, numerals, blanks, symbols, etc.) between quotation (") marks on the right side is put in the character variable specified by the left side of the equation.
When the number of characters on the right side exceeds 7, however, the first seven characters alone are put in.
Note: When clearing the equation, specify the right side of the equals as " .

General form (3) **[Character variable] = [Character variable]**
Characters stored in the character variable specified by the right side of the above equation are put in the character variable specified by the left side.

Example: (1) A = 5 ∗ 6 [ENTER] → The result of 5 ∗ 6, 30 is put in variable A.
(2) A$(27) = "USA" [ENTER]→ Characters "USA" are put in variable A$ (27).
(3) B$ = A$(9 ∗ 3) [ENTER]→ Characters stored in variable A$(27) are put in variable B$.

Example: Program that recalls fruit's names corresponding to predetermined fruit's name codes 1 through 26 put in the machine. (Indirect designation)

Key in: code (number)

Display: character (fruit's name)

(The program presupposes that fruit's names are stored in A$ through Z$.)

```
10 : INPUT  A(27)
20 : PRINT  A$ (A(27))
```

Operation

Setting to the PRO mode

[N] [E] [W] [ENTER]

10 [I] [N] [P] [U] [T] [A] [(] 27 [)] [ENTER]

20 [P] [R] [I] [N] [T]

[A] [SHFT] [$] [(] [A] [(] 27 [)] [)] [ENTER]

Contents of data memories

A	ORANGE	1
B	APPLE	2
C	BANANA	3
D	MELON	4

Operation

Setting to the RUN mode

[A] [SHFT] [$] [=] [SHFT] ["]

[O] [R] [A] [N] [G] [E] [SHFT] ["] [ENTER]

[B] [SHFT] [$] [=] [SHFT] ["]

[A] [P] [P] [L] [E] [SHFT] ["] [ENTER]

In case of data input by manual operation

Operation	Display	Note
Setting to the RUN mode		
[R] [U] [N] [ENTER]	?	
2	2_	Inputting a number
[ENTER]	APPLE	Display of corresponding fruit's name

4. Recalling the contents of variables

To recalls the contents of memories (variables), use the following form.

General from **[variable]** [ENTER]

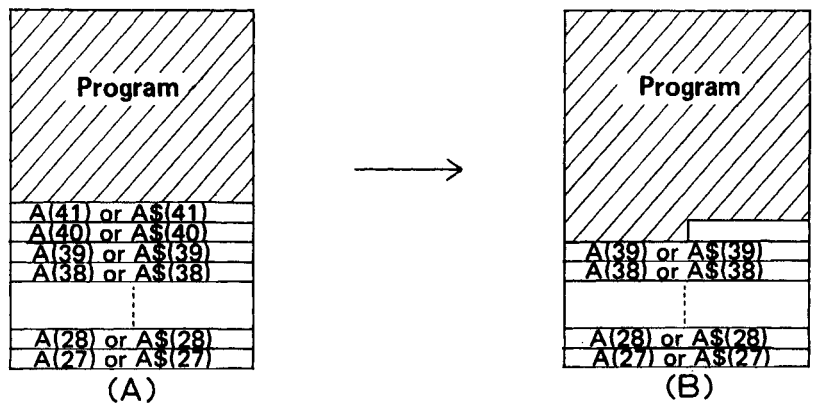
Example: When 120 and "GOOD" are stored in A and B$ respectively.

Operation	Display	Note
RUN mode [A]	A_	Designation of numerical variable
[ENTER]	120.	Display of number stored there
[B] [SHFT] [$]	B$_	Designation of character variable
[ENTER]	GOOD	Display of characters stored there

Note) If you recall the contents of cleared memories (variables) using a NEW or CLEAR command, "0." is displayed when they are specified as numerical variables and a then existing display disappears when specified as character variables. (Spacing)
Besides, if a variable cleared in the form shown in Note given for general form (2) or 3. above is recalled as specified as a numerical variable, an error also occurs (Error code: 1).

= Reference =

The program memory areas loaded with no programs are used as flexible memory. Programs can be written in the program memory unconditionally unless its capacity is exceeded. Accordingly, if program edition (insertion, deletion, correction) occurs, a change occurs in program steps, and **the number of flexible memories changes in response to that change.**
When programs are being written as shown in illustration (A) below, for example, recalling or writing the contents of flexible memories A(42) or A$(42) results in an error (Error:4) because there really exists no flexible memories corresponding to such codes.



Besides, when the arrangement of programs written (added) as shown in illustration (A) is changed to that shown in illustration (B), flexible memories corresponding to A(41) or A$(41) and A(40) or A$(40) disappear. Accordingly, if you make recall or writing for these memories, an error occurs (Error: 4).

By contrast, a string of programs stored as shown in illustration (B) is shortened as shown in illustration (A) through edition, the number of memories capable of being used as flexible memories increases in proportion to the degree of the change.
(One flexible memory corresponds to 8 program steps.)

Note) If you made recalling for increased flexible memories without writing, unexpected display might appear or an error might occur (Error: 1).

PROGRAM STATEMENTS

In the descriptions given below, [variable], [numerical variable], [character variable] and ⟨expression⟩ have individually the following meanings.

[Variable]: General name of numerical and character variables.

[Numerical variable]: General name of fixed memories defined by A through Z and dimension memories defined in the form of A ( ).

[Character variable]: General name of fixed memories defined by A$ through Z$ and dimension memories defined in the form of A$ ( ).

⟨ Expression ⟩: Operational expression composed of elements of ⟨ expression ⟩ shown on page 15, covering also [numerical variable].

1. LET statement

The variable name on the left is assigned the value of constant or expression on the right.
The PC-1211 does not require LET except when it is defined for a statement following IF statement. (For IF statement, consult page 61.)

General form (1) **LET [Numerical variable] = ⟨ Expression ⟩**

Example: LET A = 5∗3

Example:	LET A = 123	Instruction to put 123 in A (LET can be omitted as in the following example.)
	A(30) = 3∗6	Instruction to put 18 in A(30).
	A(2∗B) = C + D	Instruction to put the value of C + D in A(2∗B).

General form (2) **LET [Character variable] = "Character"**

Example: LET Z$ = "BASIC"

Characters between quotation marks are put in the character variable specified by the left side. When the length of a string of characters on the right side exceeds seven characters, however, the first seven characters alone are put in and the excess is discarded.

General form (3) **LET [Character variable] = [Character variable]**

Example: LET A$ (25) = Z$

Characters stored in the character variable specified by the right side are placed in the character variable specified by the left side.

Example:	A$ = "NON"	Instruction to place "NON" in A$.
	A$ (28) = "DATA ?"	Instruction to place "DATA" in A$ (28).
	C$ = A$	Instruction to place characters stored in A$ in C$.

- General forms (1) thru (3) can be rowed as follows by dividing them with commas (,). In this case, LET cannot be affixed behind comma (,).

  Example: 10 : LET A = 2 , B = 7 , C$ = "A = 2 B = 7"

  2 is placed in A, 7 in B and "A = 2, B = 7" in C$.

2. INPUT statement

This is an instruction to manually input data during program execution.

General form (1) **INPUT [Variable], [Variable], ···**

Example: INPUT A, B, C,

The instruction causes the computer to stop program execution and display a question mark "?". When the machine is then loaded with data and its [ENTER] key is pressed, it stores data in a specified variable.
The computer carries out this process repeatedly by the number of times that corresponds to the number of variables specified behind an INPUT statement.

General form (2) **INPUT "Character", [Variable], "Character", [Variable], ···**

Example: INPUT "A =", A, "B = ", B, ···

This form of instruction causes the computer to stop program execution and display a message instead of "?."

[Example]

Programming	Note
10 : INPUT A , B	General form (1)
20 : INPUT "C=" , C , "DATA D=" , D	General form (2)

[Execution]

Operation	Display	Note
Setting to the RUN mode		
[R] [U] [N] [ENTER]	?	Display in general form (1)
3	3_	Inputting data
[ENTER]	?	3 is placed in A. Display in general form (1)
4	4_	Inputting data
[ENTER]	C=	4 is placed in B. Display in general form (2)
5	5_	Inputting data.
[ENTER]	DATA D=	5 is placed in C. Display in general form (2)
6	6_	Inputting data
[ENTER]	>	6 is placed in D.

General from (3) **INPUT "Character"; [Variable], "Character", [Variable]···**

Example: INPUT "C = "; C , "D = " ; D ,

In general form (2), if data is put in after the display of a message, that message disappears. In general form (3), if a semicolon (;) is assigned behind "Character", message does not appear and data can be displayed following that "character."

Example

```
10 : INPUT "DATA E= " ; E
```

Operation	Display	Note
Setting to the RUN mode		
[R] [U] [N] [ENTER]	DATA E=_	A display indicating that the computer is waiting for input in general form (3).
5	DATA E=5_	Inputting data ((expression)).
[*] 6	DATA E=5*6_	
[ENTER]		30 is placed in E.

- General forms (1) to (3) can be interwoven with each other.

  Example: INPUT A , "B = " , B , "WHO ?" ; C$

- There is no limit on the length of characters between quotation marks that specify a message in general forms (2) and (3).
- You can make a correction by pressing the [CL] key in the course of inputting data.
  In general forms (1) and (2), however, pressing the [CL] key provides a display of a question mark "?" alone; in general form (3); the press provides a display of a message again.
  When you have pushed the [ENTER] key and an error has occured, the same displays as above occur, if you push the [CL] key.

Note: [Variable] specified in general forms (2) and (3) must be a fixed memory, a dimension memory that is specified by a code – such as A (30) or A (B) – that contain a fixed memory in parentheses, or an integer with no sign.
You must note that no specification cannot be made in such a form as A(A(30)) or A(5 * 9).
You must also note that indexing the [SHFT] and [↑] keys when [variable] is a character variable in general forms (1) to (3) prevents the subsequent inputs is put in a character variable.

[Skip operation]

If you press the [ENTER] key without inputting data for INPUT statements, the computer skips the remaing statements on the line and goes into the next line.

Example: PROGRAM 3 〈 Average 〉

Programming	Note
10 : "A" : CLEAR	Definition: memory is cleared.
20 : INPUT "DATA A=" ; A : B=B+A : C=C+1 : GOTO 20	Inputting data: sum Counting data number: Jump
30 : D=B/C	Computing average
40 : PRINT "AVERAGE= " ; D	(PRINT statement in general form (5))
50 : END	End

This program, a program to find averages, inputs data, sums data and counts data number repeatedly on the 20th line, and compels the machine, on the completion of data inputting, to skip the statements responsible for summing and counting, and goes into the 30th line.

〈 **Execution** 〉 Data 12, 24, 19, 23

Operation	Display	Note
Setting to the DEF mode		
SHFT A	DATA A=_	Execution begins.
12 ENTER	DATA A=_	Inputting data
24 ENTER	DATA A=_	
19 ENTER	DATA A=_	
23 ENTER	DATA A=_	
ENTER	AVERAGE= 19. 5	A skip operation occurs by pressing the ENTER key without entry of data.
ENTER	〉	

3. PRINT statement

The statement is an output instruction to display calculation results.

This instruction commands the computer to stop program execution after it has displayed information specified by this instruction.

To restart program execution, simply press the ENTER key without entry or manipulate the computer so as to execute a CONT command. You need not input anything.

General form (1) **PRINT 〈 Expression 〉**

Example: PRINT 123 + 456
PRINT A

This form commands the machine to display characters between two quotation marks from its left most one in sequence. In this case, on the length of a string of characters is available within the capacity of a line, up to 80 steps.

General from (3) **PRINT [Character variable]**

Example: PRINT A$

This form causes the computer to display the contents of [character variable] from its left-end one in sequence.

Example:

Programming	Note
10 : INPUT A$	Input instruction
20 : PRINT 5*6	General form (1)
30 : PRINT "PROGRAM A"	General form (2)
40 : PRINT A$	General form (3)

Operation	Display	Note
Setting to the RUN mode		
[R] [U] [N] [ENTER]	?	Execution of input instruction
[W] [O] [R] [L] [D] [ENTER]	30.	Inputting "WORLD"
		Display in general form (1)
[ENTER]	PROGRAM A	Display in general form (2)
[ENTER]	WORL D	Display in general form (3)

General form (4) **PRINT** { **〈 Expression 〉** / **"Character"** / **[Character variable]** } , { **〈 Expression 〉** / **"Character"** / **[Character variable]** }

Example: PRINT A, B
PRINT "A=", A

The 24-digit display is divided into two sections: right and left 12-digit sections. The right section displays the first set of 〈 expression 〉, "character" and "character variable" ahead of comma (,), and the left section another set.
The value of 〈 expression 〉 is displayed in 12 digits, being shifted to the right and a display of characters occurs in 12 digits from the left end.
When the whole numerical data cannot be displayed in 12 digits, the less significant ones of decimal digits are cut off, and when the length of a string of characters exceeds 12 digits, the first 12 digits alone are displayed.

Example: Polar coordinates → rectangular coordinates program
The program converts polar coordinates (r, θ), if they are entered in the machine, into rectangular coordinates (x, y).

```
10 : INPUT R, C
20 : X=R*COS C : Y=R*SIN C
30 : PRINT X, Y
40 : END
```

Operation	Display	Note
Setting to the RUN mode DEG		
[R] [U] [N] [ENTER]	?	
(When r = 12) 12 [ENTER]	?	
(When θ = 30°) 30 [ENTER]	10. 39230485 (Value of x) 6. (Value of y)	Two numerical values are displayed at the same time.

General form (5) **PRINT** { ⟨ Expression ⟩ / "Character" / [Character variable] } ; { "Character" / [Variable] } ;··· { "Character" / [Variable] }

Example: PRINT "A =" ; A ; "B = " ; B ;···
PRINT A$; B ; C$; C ; ···

The form provides a concurrent display of many information; it divides information items with semicolon (;).

Example:

Programming	Note
10 : "A" : CLEAR	
20 : INPUT "DATA=" ; A	Input instruction
30 : B=B+A : C=C+1	Sum: count of data number
40 : PRINT " TOTAL = " ; B ; " QTY = "; C	Total of sums: display of data number
50 : GOTO 20	Jumping to 20th line

Operation	Display	Note
Setting to the DEF mode		
[SHFT] [A]	DATA=_	
456 [ENTER]	TOTAL=456. QTY=1.	Display in general form (5)
[ENTER]	DATA=_	
789 [ENTER]	TOTAL=1245. QTY=2.	Display in general form (5)

- When the number of characters to be displayed exceeds 24, only the first 24 characters are displayed.

Note: The 2nd and subsequent display items can be only "character" and "variable"; the "variable" must be a fixed memory, or a dimension memory that has a fixed memory in parentheses or that is specified by an integer with no sign. [such a manner as A (30) or A$ (C)]

4. PAUSE statement

The PAUSE statement is an output instruction like a PRINT statement. But, these statements are different in function. Whereas the PRINT statement causes the machine to temporarily stop program execution after it has executed (displayed) a given instruction, the PAUSE statement compels it to display specified item for a certain period of time – about 0.85 sec. – and restart program execution automatically.

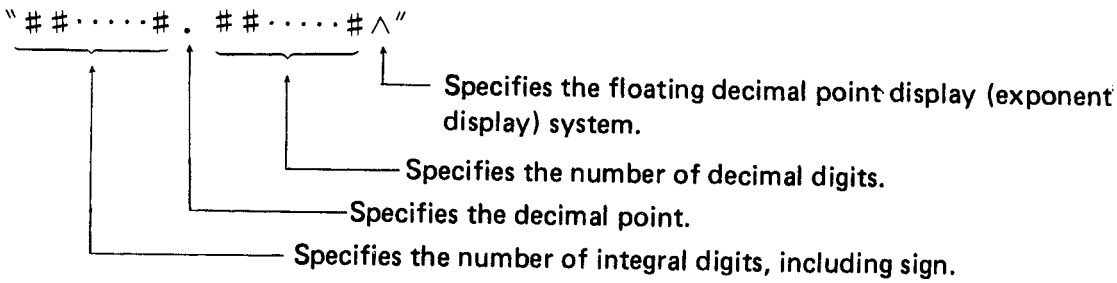
The definition form (general form) of PAUSE statement is the same with PRINT statement.

5. USING statement

The statement is an instruction to specify a display format for numerical data PRINT or PAUSE statements cause to appear on the display.

General form (1) **USING** "##…#.##…#∧" ''

This form of using statement specifies a format depending on the number of "#" between quatation marks, · and ∧.


"##·····#.##·····#∧"
Specifies the floating decimal point display (exponent display) system.
Specifies the number of decimal digits.
Specifies the decimal point.
Specifies the number of integral digits, including sign.

General form (2) **USING (Statement end)**

The term "Statement end" denotes ENTER or a colon (:).

This form commands the computer to cancel the format specification.

A display of numerical formats that occurs due to the execution of a PRINT or PAUSE statement preceded by the above form of instruction has the same format as that of numerical data in manual calculation.

Example: Displaying 123.4567891 by PRINT or PAUSE statement.

```
10 : A=123.4567891
20 : USING "Specified format"
30 : PRINT A
```

Specified format	Display
Cancel	123.4567891
#	Error (Error code: 6)
##	
###	
####	123
#####	123
####.	123.
######. #	123.4
#######. ###	123.456
####. ########	123.4567891
####. ##########	123.456789100
##. ###^	1.234E 02
####. ######^^^	1.234567E 02
###. ##########^	1.234567891E 02
. #########^	1.234567891E 02

When the floating decimal point display is specified, a 2-digit display is always assured regardless of how many integral digits you specify. Besides, the number of specified "^" does not affect the display at all.

- When numerical data cannot be displayed in a specified format, an error (Error code: 6) is encountered.

General form (3) (a) {PRINT / PAUSE} **USING "Format";**·····

(b) {PRINT / PAUSE} **USING;**·····

Example: PRINT USING "###.##";A

Formats can be specified (a) or canceled (b) by USING even in a PRINT or PAUSE statement. In this case the part behind USING statement is defined with a semicolon (;).

Example:

```
10:A=-123.456
20:PAUSE USING "####";A
30:PAUSE USING "####.#";"A=",A
40:PAUSE "A=",USING"####.#";A
50:PAUSE A,USING "####.##";A
60:PAUSE A;USING "####";A
70:PAUSE USING ;A
```

Operation	Display	Note
Setting to the RUN mode R U N ENTER	−123 A= −123.4 A= −123.4 −123.45 −123.45 −123.45−123 −123.456	

Note: A display of numerical data occurs, if USING is put in a PRINT or PAUSE statement of general form (4), according to the format in which the latter half numerical value of those statements is displayed. Accordingly, two numerical data are both displayed in the format specified for the latter data, even though their display format is individually specified. (This is true of 50th line in the above example.)

6. GOTO statement

The GOTO statement is an instruction to make program execution jump to an arbitrary line.

General form (1) **GOTO 〈 Expression 〉**

Example:
```
GOTO 10
GOTO 5*9
GOTO A
```

This form makes program execution jump to a line that corresponds to the value of 〈expression〉. The value of "expression" is effective in its integer part alone, being limitted to 1 through 999. Other values causes an error (Error code: 2).

Example:

```
    ⋮
20: INPUT "DATA=" ;D
    ⋮
80: IF  B>=CGOTO 130
    ⋮
100: GOTO 20
    ⋮
130: PRINT E, F
    ⋮
```

Example: Indirect jump

```
    ⋮
10: INPUT A
20: GOTO A
30: [ ]   <- A=30
40: [ ]   <- A=40
50: [ ]   <- A=50
    ⋮
```

Jumping occurs in accordance to the contents of A.

General form (2) GOTO { **"Character"** / **[Character variable]** }

Example: GOTO "AB"
GOTO A$

This form causes a jump to a label whose contents are the same as those of "character" or [character variable]. The length of "character" and label is effective up to seven characters. Accordingly, if those are composed of more than seven characters, program execution jumps to a label same in its first seven characters.

Example:

```
    :
30 : "A-1" : INPUT A$
    :
90 : IF P<>QGOTO A$ ----
    :                   |
120 : GOTO "A-1"        |
    :                   | A$=ME
    :                   |
240 : "ME" : PRINT C    |
    :                   | A$=YOU
    :                   |
300 : "YOU" : PRINT C   |
    :                   |
```

In the execution of "GOTO A$" on the 90th line, when the contents of A$ are "ME", a jump to label "ME" occurs, and if "YOU", a jump to label "YOU" occurs.

Note: No statement can follow a GOTO statement.
A line must end with a GOTO statement without exception.

7. IF statement

The IF statement is an instruction capable of deciding given conditions, giving "larger/smaller" decision, "equal" decision, "not-equal" decision, etc.

General form (1) **IF 〈 Expression 〉 Logic operator 〈 Expression 〉 Execution statement**
Logic operator: <, <=, =, >, >=, <>

If the relative expression that follows IF is effected — if the logical operation results in 1 (true) — the next execution statement (instruction) is executed, and if not — if the logical operation results in 0 (false) — the program execution skips the next execution statement, going into the following line.

Example:

```
    :
40 : IF A*B>=C PAUSE A*B : GOTO 90
50 : A=A+1
    :
90 : A=A+B
    :
```

If A * B > = C, the program execution begins with the execution of the following execution statement "PAUSE A*B".
If A * B < C, the program execution begins with the execution of "A = A + 1" on the 50th line, skipping "PAUSE A*B : GOTO 90".

Note: 1. If you want to execute LET statement directly after executing IF statement, put in LET without fail.

Example: IF B>C LET B=B+1

2. When GOTO statement follows IF statement, the former can be defined with THEN statement. The THEN statement has in this case the same duty as the GOTO statement.

Example: IF B>=C THEN 50↔IF B>=C GOTO 50

General form (2) IF 〈 Expression 〉 Execution statement

The 〈 expression 〉 is, if its value is larger than 0, judged to be true, and the following execution statement is executed.
The 〈 expression 〉 is, if its value is 0 or is smaller than 0, judged to be false, and the program execution goes into the next line, skipping the following execution statement.

Example:

```
   :
30 : IF AGOTO 80
40 : A=B*C
   :
```

If A>0, "GOTO 80" is executed.
If A≦0, "A = B*C" on the 40th line is executed.

General form (3) IF { **"Character"** / **[Character variable]** } = { **"Character"** / **[Character variable]** } **Execution statement**

Example: IF A$ = "ABC"
IF A$ = B$

The computer check that the contents of "character" or [character variable] of items 1 and 2 coincide, and executes the following execution statement, if they coincide; but, if they differ from each other, it skips that statement and moves to the next line.

Example:

```
   :
30 : IF A$= "GUARD" GOTO 100
40 : INPUT A$
   :
```

If the contents of A$ are "GUARD," "GOTO 100" is executed.
If not, "INPUT A$" on the 40th line is executed.

- When the length of "character" is more than seven-character equivalent, the first seven characters alone is effective for a decision, and the excess is neglected.

General form (4) IF [Character variable] Execution statement

If any character is stored in the [character variable], the following execution statement is put in execution, whereas, if nothing is present there, the machine skips the said statement, moving to the following line.

8. GOSUB statement, RETURN statement

When certain processing procedures are used several or more times in programming, you can set up a more efficient program by dealing those procedures as subroutines.
If GOSUB statement is read out during program execution, the machine moves to a specified line or label and executes as subroutine the programs stored on that line and the subsequent.
After that, ouring to RETURN statement, the PC-1211 returns to the statement next to GOSUB statement, and proceeds to execute instructions in the said statement and the subsequent.
A subroutine can contain other subroutines, enabling to give a depth of a maximum of 4 stages.

(1) GOSUB statement

General form (1) **GOSUB 〈 Expression 〉**

Example: GOSUB 10
GOSUB A

General form (2) **GOSUB** { **"Character"** / **[Character variable]** }

Example: GOSUB "ABC"
GOSUB A$

GOSUB statement allows quite the same use as does GOTO statement, however, the former one is capable of being followed by another statement.

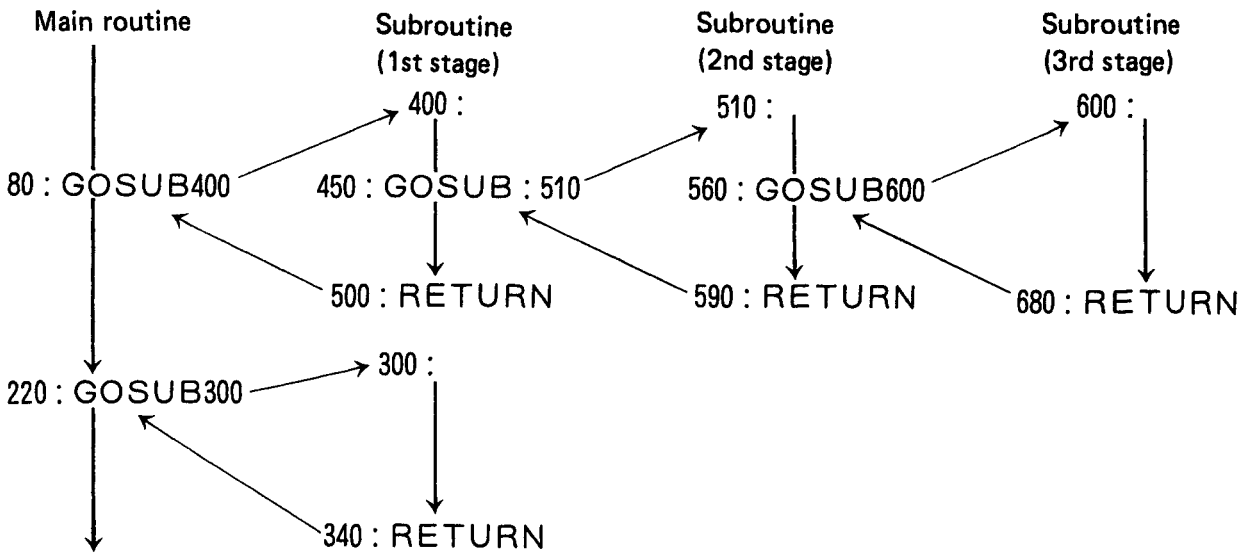
(2) RETURN statement

General form **RETURN**

RETURN statement must be present at the end of a line, not permitted to be followed by any other statements.

Note: An error (error code: 3) occurs, if a RETURN statement proceeded by no GOSUB statement is read out.

Example 1



Example 2: PROGRAM 4 〈 Approximate definite integral by Simpson's method 〉
(Refer to page 25 of APPLICATION'S MANUAL for PC-1211)

[Formula]

Compute a definite integral by using Simpson's rule.

$$S = \int_{x_0}^{x_{2p}} f(x)dx = \frac{h}{3}[(y_0 + y_{2p}) + 4(y_1 + y_3 + \cdots\cdots + y_{2p-1}) + 2(y_2 + y_4 + \cdots\cdots + y_{2p-2})]$$

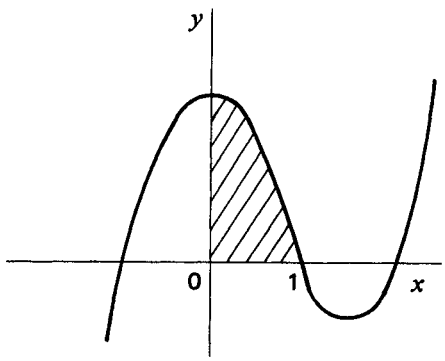
$$h = \frac{(x_{2p} - x_0)}{2p}$$

p : number of divisions

[Example]

$$y = x^3 - 2x^2 - x + 2$$
$$= ((x - 2)x - 1)x + 2$$

$$\int_0^1 y\,dx = \frac{13}{12}$$



Write in the function, as subroutine, after line 500.

How to write in (in the case of example above):

Set to PRO mode (by pressing proper MODE key).

500 Y = ((x − 2) * x − 1) * x + 2 ENTER

510 RETURN ENTER This ends writing.

Next, change to DEF mode, and execute.

Programming	Note
10: "A" : INPUT "X0= "; D, "X2P= "; E, "P= "; F 20: B = (E − D) / 2 / F 30: A = 0: X = D: GOSUB 500 40: A = Y + A: X = X + B: GOSUB 500 50: A = Y * 4 + A: X = X + B: GOSUB 500 60: A = Y + A: F = F − 1 70: IF F <> 0 GOTO 40 80: C = A * B / 3 90: BEEP 3: PRINT "ANS.", C 100: END 500: Y = ((X − 2) * X − 1) * X + 2 510: RETURN	

Operation	Display		Note
Setting to the DEF mode			
[SHFT] [A]	X0 =		
0 [ENTER]	X2P =		(x_0)
1 [ENTER]	P =		(x_{2p})
20 [ENTER]	ANS.	1.083333333	(P)

When the [ENTER] is pressed at this step, the PC-1211 starts the calculation with displaying the symbol "RUN" and about 40 seconds later, an answer will be displayed with 3 times of beep soud.

** Debugging the program enables you to better understand its execution process. Try to debug it.

9. FOR statement, NEXT statement

When you need to execute identical programs many times repeatedly, or when you need to solve a calculation equation repeatedly by replacing only the values of variables contained in it, the use of FOR-NEXT statement will provide an efficient program execution. The decision function of GOTO and IF statements is of course available.

General form (1) **FOR [Numerical variable] = ⟨ Expression 1 ⟩ TO ⟨ Expression 2 ⟩**
NEXT [Numerical variable]

Example:
```
FOR A = 1 TO 26
  :
NEXT A
```

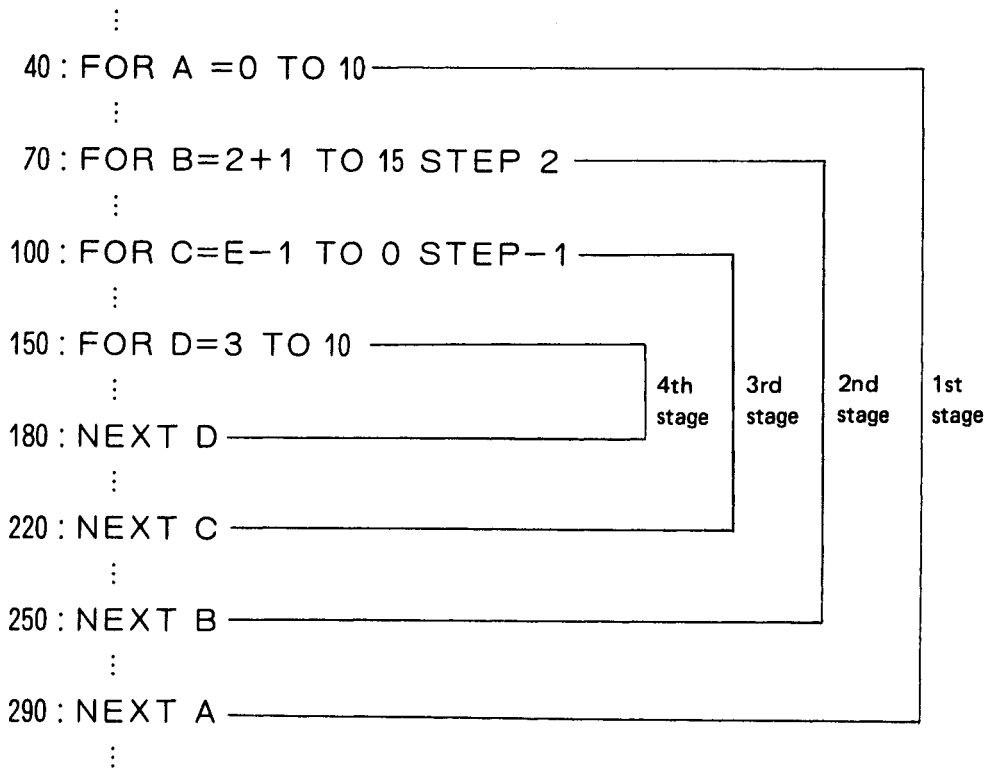
The FOR and NEXT statements defined with identical numerical variables are used so as to make a pair, and instructions between these statements are repeatedly executed by the specified number of times.
In the first execution, the value of ⟨ expression 1 ⟩ is stored as initial value in a specified numerical valuable.
The computer, if it encounters the NEXT statement, adds an increment of 1 to the numerical variable and execute the instructions present between the FOR and NEXT statements until the contents of the said numerical variable is equal to or more than the final value that correspoinds to the value of ⟨ expression 2 ⟩.

General form (2) **FOR [Numerical variable] = ⟨ Expression 1 ⟩ TO ⟨ Expression 2 ⟩ STEP ⟨ Expression 3 ⟩**
NEXT [Numerical variable]

Example:
```
FOR A = 1 TO 26 STEP 2
  :
NEXT A
```

The form puts the machine in the same operation as does general form (1). However, this form can set an increment (or decrement) with the value of ⟨expression 3⟩.
In program execution, a set increment is added to the [numerical variable] every execution. If the value of ⟨ expression 3 ⟩ is negative, program execution occurs repeatedly until the value of the [numerical variable] is equal to or smaller than the value ⟨ expression 2 ⟩ exhibits.

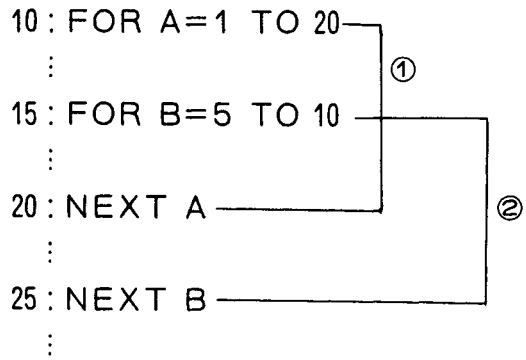
- For the values of ⟨ expression 2 ⟩ and ⟨ expression 3 ⟩ their integers alone are effective, limited to less than three digits. When the value of ⟨ expression 3 ⟩ is 0 in its integral part, an error (error 1) occurs.
- A FOR-NEXT statement can be given a depth of a maximum of 4 stages.


40 : FOR A =0 TO 10
70 : FOR B=2+1 TO 15 STEP 2
100 : FOR C=E−1 TO 0 STEP−1
150 : FOR D=3 TO 10
180 : NEXT D
220 : NEXT C
250 : NEXT B
290 : NEXT A
4th stage
3rd stage
2nd stage
1st stage

Note: For a FOR-NEXT statement, it makes processing faster to specify memory 23 (W) and the subsequent rather than to specify memories that precede that memory.

In the following cases, an error occurs during program execution.

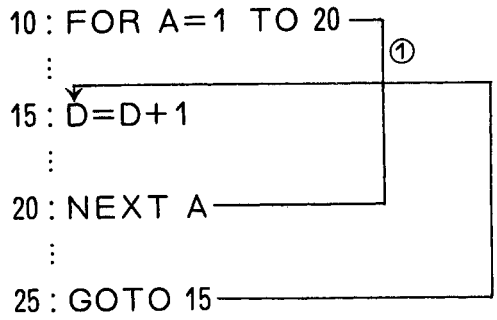
⟨ Crossing ⟩



Crossing of (1) and (2)

In this case, although FOR-NEXT statement of ① is executed, an error (error 4) occurs in the execution of line 25 because "FOR B = 5 TO 10" on line 15 is neglected.

〈 **Joining in mid course** 〉



Line 25 joins line 15 with the loop of ①

The result is that program execution moves from line 25 to 15, and gives rise to an error (error 4) as soon as it reaches line 20.

Example: **Program 5 〈 Decision of order 〉**

Programming	Note
10 : "A" CLEAR :A=5	Label A
20 : INPUT "DATA=" ; D	Input program
30 : IF D=99GOTO 50	
40 : A (A) =D : A=A+1 : GOTO 20	
50 : FOR B=5TO A-1	② ①
60 : FOR C=B+1TO A-1	
70 : IF A (B) > =A (C) GOTO 110	
80 : D=A (B)	
90 : A (B) =A (C)	
100 : A (C) =D	
110 : NEXT C	
120 : NEXT B	
130 : "B" FOR B=5TO A-1	Label B
140 : BEEP 2 : PRINT B-4 , A (B)	③ Output program
150 : NEXT B	
160 : BEEP 5 : END	

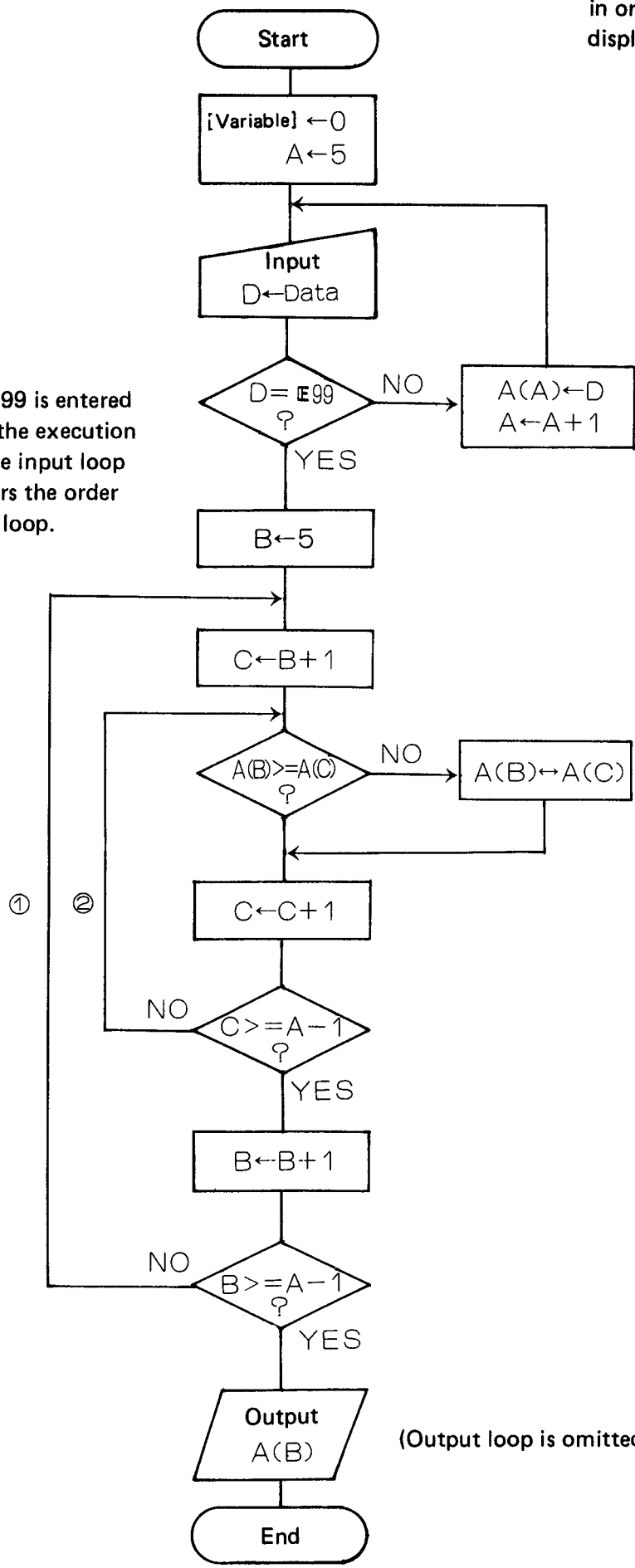
①, ② and ③ indicate the loops of FOR-NEXT statement.

** Program debugging enables you to easily understand the process of program execution and the machine movement caused by a FOR-NEXT statement.

〈 **Flow chart** 〉

Decision of order

Input numerical data are arranged in order of their magnitudes and displayed in sequence.



Preparation

Input loop

Note:
When **E** 99 is entered as data, the execution leaves the input loop and enters the order decision loop.

(Output loop is omitted.)

10. STOP statement

This statement is an instruction to temporarily stop program execution.

General form **STOP**

The computer displays, when it has executed the STOP statement, BREAK message together with line number, and stops program execution.
The manual operation is then possible. To restart the program execution, have the computer execute a CONT command.

Example:

```
100 : C=3*7
200 : STOP
300 : PRINT "C=" ; C
      :
```

It a STOP statement is executed on line 200, the computer displays a BREAK message as shown below.

Operation	Display	Note
	BREAK AT 200	Display of BREAK message
C	C_	When program execution stops due to STOP statement, the manual calculation is possible.
ENTER	21.	
4 * 8 ENTER	32.	
C O N T	CONT_	CONT command restarts the program execution.
ENTER	C=21.	

11. END statement

The END statement is an instruction to end program execution.

General form **END**

The execution of this instruction causes the machine to end program execution and display the prompt symbol.

12. BEEP statement

The statement is an instruction to force the computer to beep.

General form **BEEP 〈 Expression 〉**

Example: BEEP 10
BEEP A

The computer beeps by the number of times that corresponds to the value of 〈 expression 〉.
That value is effective only in its positive integral part.

13. CLEAR statement

The statement is an instruction to clear all data memories: fixed and flexible memories.

General form **CLEAR**

- Program and reserve memories are protected.

14. DEGREE, RADIAN, GRAD statements

These statements are instructions to specify the unit of angle for the input of trigonometric functions (SIN, COS, TAN) and for the output of inverse trigonometric functions (ASN, ACS, ATN).

(1) DEGREE

The statement sets the unit of angle to degree.

General form **DEGREE**

(2) RADIAN

The statement sets the unit of angle to radian.

General form **RADIAN**

(3) GRAD

The statement sets the unit of angle to gradian.

General form **GRAD**

$$90^\circ = \frac{\pi}{2} \text{ [rad]} = 100^g$$

15. AREAD statement

The statement is an instruction to automatically store numerical value, the value of ⟨ expression ⟩ or "character" in a specified variable that has been displayed before the start of program execution. This instruction performs its duty when the computer first encounters, in the DEF mode, the AREAD statement in the execution of a definable program.

General form **AREAD [Variable]**

Example:
```
AREAD A
AREAD A$
```

This instruction is skipped, if not present at the head of a definable program.

Example: Compound-interest computation program.

Programming	Note
10 : "A" : AREAD I	Inputting of interest rate (%)
20 : I = I / 100	
30 : END	
40 : "S" : AREAD N	Inputting of term
50 : END	
60 : "D" : AREAD P	Inputting of principal
70 : END	
80 : "F" : F = P * (1 + I) ^ N	Computation of principal plus interest
90 : PRINT F	Display of principal plus interes interest
100 : END	

Operation	Display	Note
Setting to the DEF mode		
6.8 [SHFT] [A]	)	Interest rate 6.8%
4 [SHFT] [S]	)	Term 4 years
5000 [SHFT] [D]	)	Principal DLS 5,000
[SHFT] [F]	6505.115547	Principal plus interest
5 [SHFT] [S]	)	Term changed to 5 years
[SHFT] [F]	6947.463404	Principal plus interest

Note: A specified variable (memory) is loaded with the contents of its right-hand one for a display in general form (4) of PRINT statement or with the contents its first one for a display in general form (5).

16. REM statement

The statement, not an execution statement, is an instruction to specify a note in a program for easier future checking. The program execution skips a statement — note — that follows this instruction, going into the following line.

General form **REM 〈 Note 〉**

Use this statement when you want to insert notes between programs for dividing them clearly so as to give no affects on their execution.

Example:

```
    :
80 : REM * OUTPUT  PROGRAM *
    :
```

COMMAND STATEMENTS

The PC-1211 can process, in addition to program execution statements, instructions capable of starting the program execution or displaying program contents, each being called a command. These commands must end, without execption, with [ENTER] , and are executed as soon as the [ENTER] key is pressed.

1. RUN command

The RUN command, effective in the RUN or DEF mode alone, specifies the start of program execution.

General form (1) **RUN** [ENTER]

This form starts the program execution at the head line of a program.

General form (2) **RUN 〈 Expression 〉** [ENTER]

Example: RUN 30 [ENTER]

This form starts the program execution at the line specified by the value of 〈 expression 〉. That value is effective in its integeral part, limited to 1 through 999.

Example: [R] [U] [N] 10 [ENTER] Program execution begins at line 10.

General form (3) RUN { "Character" / [Character variable] } [ENTER]

Example: RUN "ABC"

This form starts the program execution at the line given a label the same with the character stored in "character" or [character variable].
When the length of "character" and label exceeds seven characters equivalent, then first seven characters are checked for coincidence and the excess is skipped.

Example: [R] [U] [N] [SHFT] [↑↑] [P] [R] [O] [−] [1] [SHFT] [↑↑] [ENTER]

The program execution starts at the line labeled "PRO−1".

- If given this command when not programmed, the computer provides a prompt (>) display at once.
- When the specified line does not exist in general form (2) or when the specified label does not exist in general form (3), an error (error 2) occurs.

2. DEBUG command

The command, effective in the RUN or DEF mode, can be used in the same form as RUN command. However, if a program is executed with this command, the machine displays the line number of executed line before it moves from that line to other line, put in the break condition (temporary stop).
If the [I] key is then pushed, the machine advances to the next execution line and displays its line number before leaving that line, going into break again. (Such a line-by-line execution is called debugging.)

Note: For program debugging, consult page 37.

- The general form of this command can be defined in the same manner as that of RUN statement.

3. CONT command

The CONT command, effective in the RUN or DEF mode, clears the temporary stop of program execution to make the machine continue it in the usual state.

General form CONT [ENTER]

The temporary stop of program execution can be described below.

(1) Break condition
- ① Temporary stop due to STOP statement during program execution.
- ② Temporary stop due to the press of the [ON] key during program execution.
- ③ Temporary stop accompanied by a display of a line number in debugging.

(2) After the execution of a PRINT statement (while specified contents are displayed)

Example:

```
10 : A=0
20 : FOR  B=1 TO  3
30 : A=A+B : PAUSE  B, A
40 : NEXT  B
50 : END
```

Operation	Display	Note
Setting to the RUN mode		
D E B U G	DEBUG_	
ENTER	10:	
↓	20:	
↓	1. 1.	
	30:	
↓	40:	
C O N T	CONT_	Inputting of a CONT command
ENTER	2. 3.	Execution of the CONT command (Reset of temporary stop)
	3. 6.	

4. LIST command

This command lists a program, being effective in the PRO mode.

General form (1) **LIST** ENTER

The form puts the head line of a program on the display.

General form (2) **LIST 〈 Expression 〉** ENTER

Example: LIST 10

This form compels the computer to display a programs on the line specified by the value of 〈 expression 〉.
That value is operative in its positive integer part, limited to 1 through 999.

General form (3) **LIST** { **"Character"** / **[Character variable]** } ENTER

Example: LIST "ABC"

This form commands the computer to display programs on the lines given the same labels with characters stored in "character" or [character variable].
When the length of "character" and label exceeds seven-character equivalent, their first seven characters are check for coincidence and the excess is skipped.

Example:

```
10:A=0
20:FOR B=1TO 3
30:A=A+B:PAUSE B, A
40:NEXT B
50:END
```

Operation	Display	Note
Setting to the PRO mode		
[L] [I] [S] [T]	L I S T_	
[ENTER]	1 0 : A=0	Display of the head line
[L] [I] [S] [T] 3 0	L I S T 3 0_	
[ENTER]	3 0 : A=A+B : PAUSE B , A	Display of line 30

- Recall an arbitrary line by LIST and the preceding and following lines by the aid of the [↑] or [↓] key. This will provide an efficient program checking.
- When programs on a specified line cannot be displayed in 24 digits, they are displayed from first one in sequence as much as possible. At this time they are recalled by shifting the cursor by the aid of the [►] key.
- When the specified line does not exist in general form (2) or when the specified label does not exist in general form (3), an error (error code: 2) occurs.

5. NEW command

The command clears programs, reserve programs, and data.

General form **NEW** [ENTER]

(1) DEF, RUN and PRO modes

The execution of a NEW command in these modes clears all program and data memories.

(2) RESERVE mode

The execution of a NEW command in this mode clears all reserve memories.

- The prompt symbol appears after the execution of this command.

6. MEM command

This command forces the computer to display the number of program steps and flexible memories, thereby showing the number of empty program memories.

General form **MEM** [ENTER]

This command is operative in all the modes.

Example: The execution of a MEM command with program memories all cleared.

Operation	Display	Note
PRO mode		
[N] [E] [W] [ENTER]	>	
[M] [E] [M] [ENTER]	1 4 2 4 S T E P S 1 7 8 M E M O R I E S	※

※ The displays shows that programming is further possible up to 1424 steps or that a maximum of 178 memories can be used as flexible memories.

One flexible memory corresponds to 8 steps of program.

Accordingly, if the empty program memory is loaded with a program of within 8 steps, the number of flexible memories are reduced by one, and if loaded with a program of 9 through 16 steps, that number is reduced by two. In this way, if a program is entered, the number of flexible memories is reduced corresponding to the number of its steps.

- The displayed number of memories does not include that of fixed memories.

ERROR

If you execute the contents the computer cannot process, such as grammer not defined by it or unreasonable operational specification, an error will occur. The computer displays an error code, stopping the execution.

Example 1

Operation	Display
RUN mode	
5 [+] [*] 3	5+*3_
[ENTER]	1. . . . .
[►]	5+*3

In this example, * causes a grammatical error, which ferces the error symbol to show up on the display.
If the [►] or [◄] key is then pressed, the contents of the input are displayed with the cursor staying at the position of * where the error occurred.

Example 2

```
    ⋮
30 : A=5+*3
    ⋮
```

Operation	Display
RUN mode	Line number ↓ 30 : Error code ↓ 1. . . . .
[↥]	30 : A=5+*3
	> Error position ↑

Example 2 above shows the case where an error occurred on line 30.
In this case, the computer displays, in addition to an error code, the line number of a line on which the error occurred.
If the [↥] key is then pressed, the area where the error occurred is displayed, and the display continues during that key is pressed. (If the said key is released, the prompt symbol (>) appears.)
To correct the error area in program operation, press the [MODE] key in succession to choose the PRO mode and press the [↥] or [↧] key to display the corresponding program for correction.
To clear an error in manual operation, use the [ON] or [CL] key. For other errors than the step -over of program memory and error code 5, the [►] or [◄] key is also available.
An error encountered in program execution can be cleared by the aid of the [ON], [CL] or [↥] key. (An error encountered during the execution of a CHAIN statement cannot be cleared by means of the [↥] key.)

〈 **Error code and the nature of errors** 〉

Error code	Nature	Description
1	• Grammatical error • Operational error • Error in memory specification	• Occurs when the absolute value of a calculation result exceeds 1×10^{100}, or when the divisor is 0. • Occurs when memories to which numerical values are assigned are specified as character variables or in the reverse case.
2	Line error	• Occurs when lines and labels specified by GOTO, GOSUB, RUN, DEBUG or LIST statement do not exist.
3	Error in depth	• Occurs when the depth exceeds 4 stages in a GOSUB statement or FOR-NEXT statement. • Occurs when you try to execute a RETURN statement despite the absence of a preceding GOSUB statement. • Occurs when you try to execute a NEXT statement despite the absence of its mating FOR statement.
4	Step-over error	• Occurs when you try to write programs larger in the number of steps than the remaining program memories. • Occurs when you try to write reserve programs larger in the number of steps than the remaining reserve memories. • Occurs when existing dimension memories are specified despite the absence of it.
5	Control error of magnetic tape	Occurs when an error arises in the execution of magnetic tape control instruction. (Verify error, check sum error, etc.)
6	Error in format	Occurs when a display of numerical date with PRINT or PAUSE statement in program execution does not occur in a specified format.

MAGNETIC TAPE CONTROL INSTRUCTIONS

The PC-1211 can use a tape recorder as its external storage unit by connecting them through an optional cassette interface CE-121. This will enable you to record programs, reserve programs and the contents of data memories stored in the computer on a magnetic tape (cassette tape) or, in reverse, to read them out of a tape and further to compare the contents of a tape with the contents loaded on the computer.
Giving file names to all recordings, including programs, reserve programs and data, the computer provides an automatic search when reading them out.
For the operation of a tape recorder to be connected, consult page 86, to execute the instructions described below.

1. CSAVE (Cassette save) statement

The statement is an instruction to record programs or reserve programs on a magnetic tape, **being executable only by manual operation.**

General form **CSAVE "File name"** [ENTER]

If the length of file name is more than seven-character equivalent, the excess is skipped.
This is true of all magnetic tape control instructions.

(1) DEF, RUN and PRO modes

The instruction commands the computer to record first specified file names on a magnetic tape and then the whole program stored in the program memory thereon.
However, when the memory is loaded with no programs, the prompt symbol appears on the display at once.

Example: PRO mode
CSAVE "PROG.−1" [ENTER]
"PROG.−1" is recorded as file name on a tape and all the lines that carry programs are recorded.

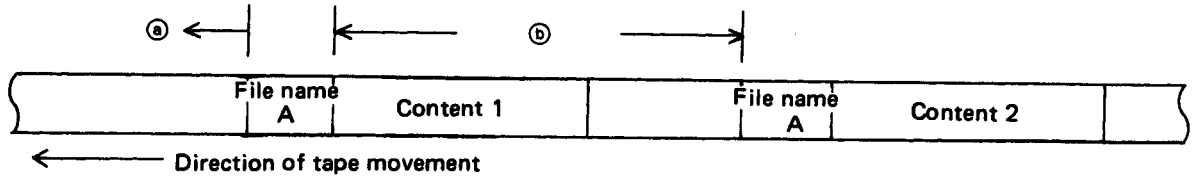
(2) RESERVE mode

The instruction forces the tape recorder to record specified file names and then all reserve programs.
However, when the memory is loaded with no reserve programs, the prompt symbol shows up.

Example: RESERVE mode
CSAVE "RESRV−1" [ENTER]
"RESRV−1" is recorded as file name on a tape and all reserve programs are recorded.

[You are requested to check, by executing a CLOAD? statement mentioned later after the execution of a CSAVE statement, that programs and datas have been securely recorded.]

Note: You must avoid, recording programs given the same file name − but different in contents − on the same side of the same tape, otherwise the reading (transferring) of wrong contents may occur in the execution of a CLOAD or CHAIN statement.



If two same file names – A in the above example – are present, a CLOAD or CHAIN statement must be executed, in order to transfer content 2, for example, after the tape has reached range ⓑ, because if the tape is started at ⓐ range for transferring content 2, content 1 is transferred instead.

Beside, of a new item is recorded overlapping, even though slightly, the previous recording, the previous one is partly destroyed, resulting in an error when it is transferred to the computer. Avoid such a overlap by all means.

2. CLOAD (Cassette Load) statement

This statement is an instruction to transfer (load on) programs or reserve programs from a magnetic tape to the computer, **being executable by manual operation alone.**
When transferring those programs, manipulate the tape recorder to rewind the tape so that the area in which they are recorded reaches the tape recorder head, before executing this instruction.

General form **CLOAD "File name"** [ENTER]

(1) DEF, RUN and PRO modes

This instruction makes an automatic reference of specified file names and transfers the corresponding programs from the magnetic tape to the computer.

Example: PRO mode
CLOAD "PROG.–1" [ENTER]
A program on the magnetic tape whose file name is "PROG.–1" is found out and transferred to the program memory.

(2) RESERVE mode

This instruction makes an automatic reference of specified file names and transfers the corresponding reserve programs from the magnetic tape to the computer.

Example: RESERVE mode
CLOAD "RESRV–1" [ENTER]
A reserve program on the tape whose file name is "RESRV–1" is found out and transferred to the reserve memory.

Note 1) The computer cannot decide whether a certain file name refers to a program or reserve program. Therefore, an improper mode selection leads to an improper transfer: reserve programs to the program memory or programs to the reserve memory.

2) If file names you want to find out are not present on a magnetic tape, **the computer continues to search the absent file names even after the tape has come to an end.** (In this case, cancel the instruction by pressing the [ON] key.)
This is true of CLOAD?, CHAIN, and INPUT # statements described later.

3) If an error is encountered during the transfer of programs, the program memory alone is ineffective. This is also true of a CHAIN statement discussed later.

3. CLOAD? (Cassette Load?) statement

The statement is an instruction to check the contents of the program or reserve memory inside the computer with the recordings on a magnetic tape that have specified file names, **being executable by manual operation alone.**
If the above check reveals the disagreement, an error (error code: 5) occurs.

(When making the above check, manipulate the tape recorder to rewind the tape so that the area to be checked reaches the tape recorder head, before executing this instruction.)

General form **CLOAD? "File name"** ENTER

(1) DEF, RUN and PRO modes

This instruction checks the contents of program memory with the recorded programs on a magnetic tape that have specified file names.

(2) RESERVE mode

This instruction checks the contents of reserve memory with the recorded programs on a magnetic tape that have specified file names.

- When the program memory is loaded with nothing in the DEF, RUN or PRO mode or when the reserve memory is loaded with nothing in the RESERVE mode, the execution of a CLOAD? statement causes a display of the prompt symbol at once.

4. CHAIN statement

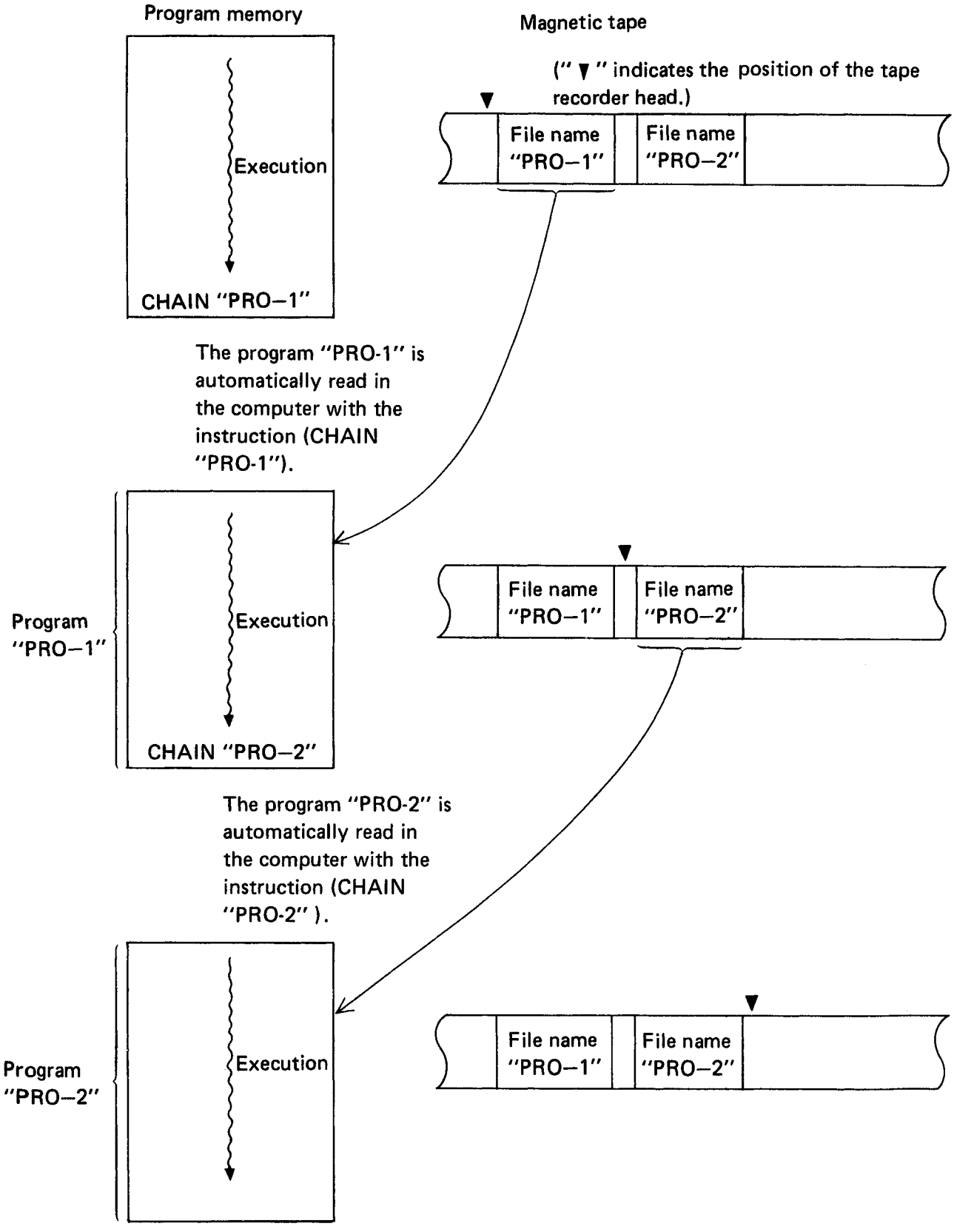
The CHAIN statement is a program execution instruction. Encountering this instruction in program execution, the machine automatically searches programs recorded on a magnetic tape that have specified file names and transfers those programs to its program memory.
And it starts the execution of the transferred programs at the line specified by the CHAIN statement.

In other words, if this instruction is used, even programs long enough to exceed the capacity of the computer's program memory can be, if recorded on a magnetic tape as divided so as to be completely stored in the said program memory, read out to the computer in succession to be executed.
(Manipulate the tape recorder to rewind the tape and then execute specified programs.)

General form (1) **CHAIN "File name"**

Example: CHAIN "PRO–1"

The instruction compels the machine to execute a transferred program from its beginning.


Program memory
Magnetic tape
(" ▼ " indicates the position of the tape recorder head.)
Execution
CHAIN "PRO–1"
File name "PRO–1"
File name "PRO–2"
The program "PRO-1" is automatically read in the computer with the instruction (CHAIN "PRO-1").
Program "PRO–1"
Execution
CHAIN "PRO–2"
File name "PRO–1"
File name "PRO–2"
The program "PRO-2" is automatically read in the computer with the instruction (CHAIN "PRO-2").
Program "PRO–2"
Execution
File name "PRO–1"
File name "PRO–2"

If each program is arranged to end with a CHAIN statement as shown above, a new program can be automatically transferred from the tape and executed in succession every time the preceding program is completely executed.

General form (2) **CHAIN "File name", ⟨ Expression ⟩**

Example: CHAIN "PRO–1", 30

This form starts the execution at the line specified by the value of ⟨ expression ⟩ contained in a transferred program.
The value of ⟨ expression ⟩ is effective only in its integer part, limited to positive numbers from 1 through 999.

General form (3) **CHAIN "File name", { "Character" / [Character variable] }**

Example: CHAIN "PRO–1", "A"

This form starts the execution at the eine given the same label with the contents of "character" or [character variable] contained in a transferred program.
At this time, the length of "character" or label is effective up to seven characters, and the excess is neglected.

Example:
```
    :
100 : CHAIN "ABC"
```
A program given file name "ABC" is transferred from the magnetic tape to the program memory, and is executed from its beginning.

Example:
```
    :
200 : CHAIN "XYZ", 10
```
A program given file name "XYZ" is transferred from the magnetic tape to the program memory and executed from its line 10.

5. PRINT # (Print cross-hatch) statement

The statement is an instruction to record the contents of data memories on a magnetic tape, **executable both by program and manual operation.** (Executable in the DEF and RUN modes)

General form (1) **PRINT # "File name"**

Example: PRINT # "DATA 1"

This form commands the computer to record "file name" and then record all the contents of data memories in sequence starting with fixed memory A (or A$).

General form (2) **PRINT # "File name"; [Label of variable]**

Example: PRINT # "DATA 1"; A(5)

This form commands to first record file names on a magnetic tape and then record the contents of specified data memory and the subsequent.
The [label of variable] is specified by characters A through Z or in the form of A ( ). In the latter case, however, material in parentheses is limited to positive integers from 1 to 204, or to fixed memories. (If program memories loaded with program are specified as flexible memories, an error occurs.)
This method of specification is also applied to an INPUT # statement described below.

A PRINT # statement must end with [ENTER] without exception, when executed by manual operation.

Example: Execution through manual operation
```
PRINT # "DATA–1"  [ENTER]
```
"DATA–1" is recorded as file name, and the contents of data memory no. 1 (memory A or A$) and the subsequent that can be specified as flexible memories are all recorded in sequence.

Example: Execution by program
```
    :
150 : PRINT # "DATA–1"; A(26)
    :
```
"DATA–1" is recorded as file name, and the contents of data memory no. 26 (memory Z or Z$) and the following are recorded in sequence.

6. INPUT # (Input cross-hatch) statement

The statement is an instruction to transfer data recorded on a tape to the data memory area of the computer, **executable both through program or manual operation.** (Executable in the DEF or RUN mode)
(Rewind the tape by manipulating the tape recorder before executing this instruction.)

General form (1) **INPUT # "File name"**

Example: INPUT # "DATA 1"

This form commands the computer to automatically search specified file names and loads data memory no. 1 (A or A$) and the following of the computer in sequence with the corresponding recorded data.

General form (2) **INPUT # "File name"; [Label of variable]**

Example: INPUT #"DATA 1"; A(5)

The form compels the computer to search specified file names automatically and loads the data memory specified by [label of variable] and the following, in sequence, with recorded data corresponding to the specified file names.

Example: Execution through manual operation

INPUT # "DATA–1" [ENTER]

Recorded data whose file name is "DATA–1" is put in data memory no. 1 and the subsequent in sequence.

Example: Execution through program

```
    :
50 : INPUT # "DATA–1"; A$(26)
    :
```

Recorded data whose file name is "DATA–1" is put in date memory no. 26 and the subsequent in sequence.

- The computer can distinguish and transfer file names recorded as program and as data even they are identical.

Note: If the number of recorded data is smaller than that of data memories to be loaded with them, the execution activated by the INPUT # statement ends as soon as all the data are transferred.
In the reverse case of the above, data are transferred until all the data memories are loaded, when the execution comes to an end.

CONNECTION OF CASSETTE INTERFACE (CE-121)

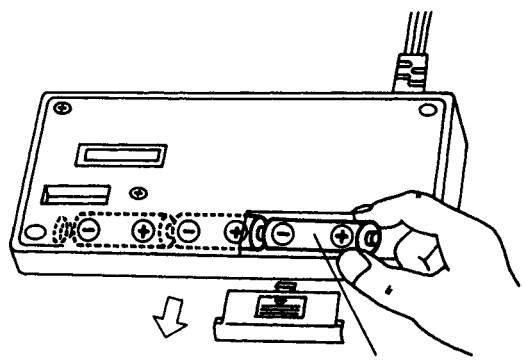
The cassette interface (CE-121) is optionally available for this computer. This unit provides the connection between the computer and an external tape recorder, thereby enabling to record and store the programs and data in magnetic tapes used for tape recorders.
These programs and data can be read in from magnetic tapes to the computer at any time, which saves you the trouble of individually keying them in.

Sequence of explanations

1. Battery replacement
2. Connecting the PC-1211 and the CE-121
3. Connecting the CE-121 and a tape recorder
   (1) Requirements for usable tape recorders
   (2) Connecting order of the CE-121 and a tape recorder

1. Battery replacement

The batteries of CE-121 are used to control the operation of the tape recorder (start/stop of tape motion). Consumed batteries make impossible remote control of the tape motion in recording or loading, even if the black plug is being connect into the REMOTE (REM) jack on the tape recorder. In this case, replace the used dry batteries with new ones in no time.



UM-3(E) (AA or R6) x 3

Note:
- After one year from the date of purchase, check the battery condition sometimes to avoid possible damage caused by battery leakage.
- Always replace all 3 batteries at the same time.
- Keeping a dead battery in the battery compartment may result in damage to the computer due to solvent leakage of the battery. Remove a dead battery promptly.

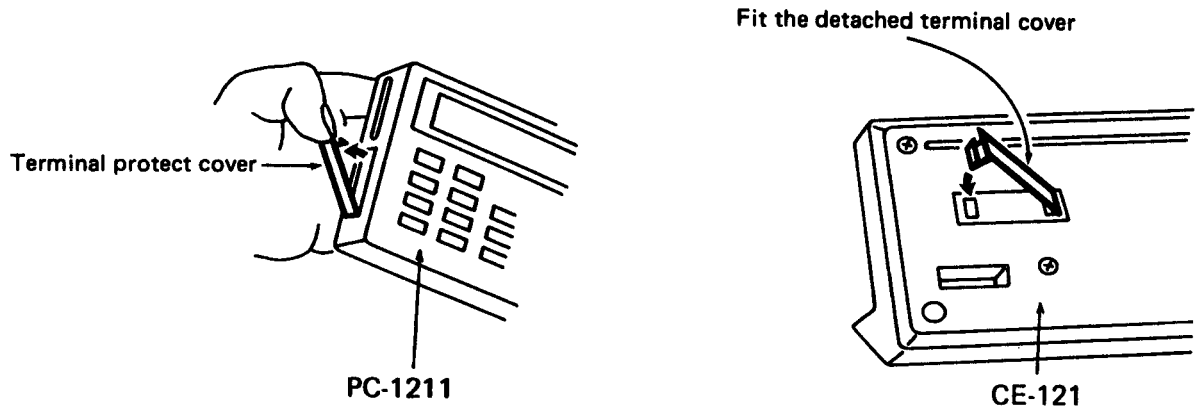
2. Connecting the PC-1211 with the CE-121

Make connection in the following procedure.

① **Turn the PC-1211 off by pushing the [OFF] key.**

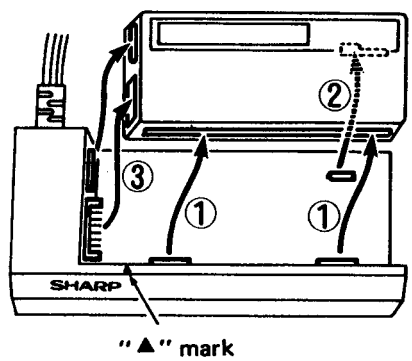
② Remove the terminal protect cover on the left hand side of the PC-1211, and attach it on the specified position on the bottom of the cassette interface.

Fig. 1



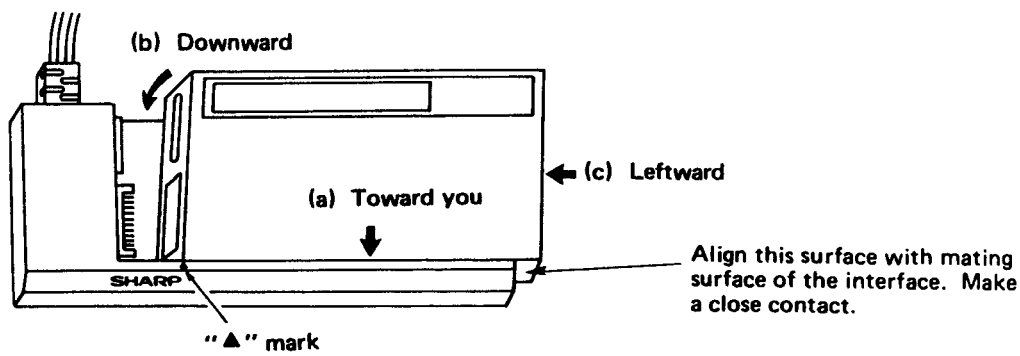
③ Fit the projecting parts on the cassette interface in the specified grooves of the computer in numerical order as shown below. See carefully the arrow marks.

Fig. 2



④ Place the machine on the cassette interface, matching the left lower angle of the computer with the triangular mark (▲).

Fig. 3



⑤ When the part marked with the arrow ② in Fig. 2 is not engaged, shift the computer slightly to right or left.

⑥ After setting the above part, shift the computer leftward so that the parts (connecting terminals of computer and interface) marked with the arrow ③ are fitted together.

- The connecting terminal should be fitted in securely. Never force it in, however.

Note: Before attaching or detaching the computer to or from the interface, be sure to turn off the computer with the [OFF] key.
In the event that the computer is connected to or disconnected from the interface with its power switch at ON, all its keys may be inoperative. In case you encounter such a situation, press the ALL RESET switch on the bottom of the PC-1211 to reset it. This will clear the entire computer. Operate it from the beginning again.

3. Connecting the CE-121 with a tape recorder

(1) Conditions required for the connection of a tape recorder unit

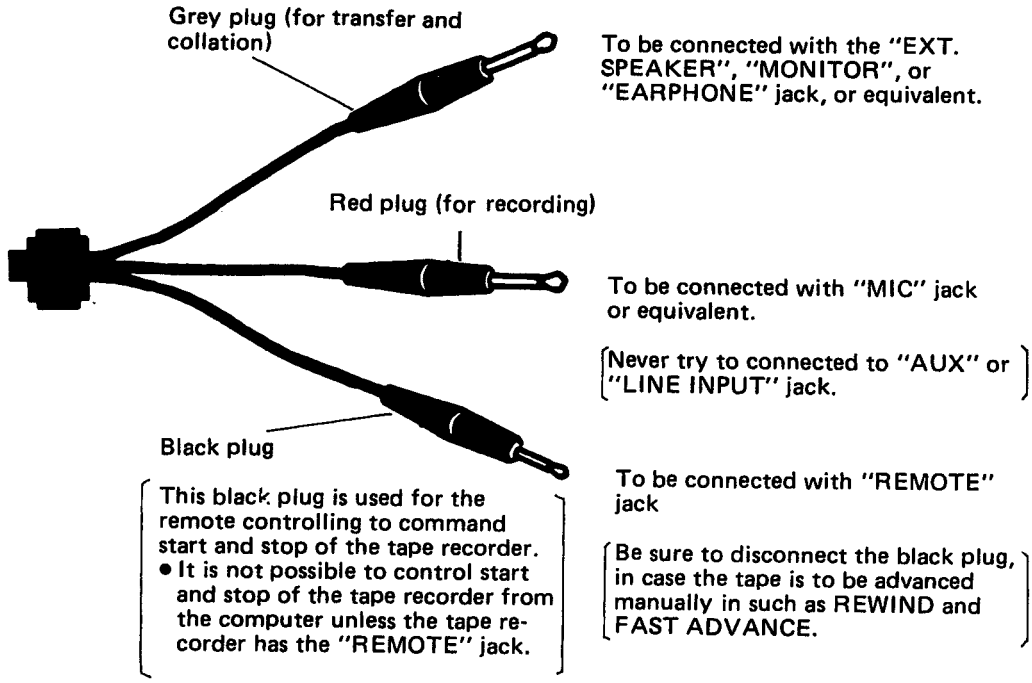
Basically, the following conditions are required to make the PC-1211 connected with a tape recorder unit using the CE-121.

Item	Conditions
Type	Tape recorder or tape deck using cassette tape, micro-cassette tape, or open reel tape.
Input jack • Input impedance • Minimum input level	Should have "MIC" jack or equivalent (mini-jack). Caution: Use of "AUX" jack is not allowed. The above jack should have impedance of 200Ω ~1KΩ or be designed for a low impedance. Below 3mV or −50dB.
Output jack • Output impedance • Output level	Should have "EXT. SPEAKER", "MONITOR", "EARPHONE" jack or equivalent (mini-plug). The above jack should have impedance of below 10Ω. Should be above 1V (practical maximum output being above 100mW).
Distortion factor,-including phase distortion	Should be within 15% (same phase and within a range of 2KHz thru 4KHz).
Wow and flatter	0.3%, maximum. NOTE: Wow and flatter may be increased when the tape recorder is subjected to an vibration and impact.
Others	No extraordinary fluctuation should be seen in motor rotation.

- In case the mini-plug of the CE-121 is not suitable for the jack of the tape recorder, use the special line conversion plug available on the market.

NOTE:
Some of tape recorders may reject connection due to different specifications. Or those tape recorders having distortion, increased noise, and power deterioration after long years of use may not show satisfactory results owing to change in their electrical characteristics.

(2) About the plug connection of CE-121


Grey plug (for transfer and collation)
To be connected with the "EXT. SPEAKER", "MONITOR", or "EARPHONE" jack, or equivalent.
Red plug (for recording)
To be connected with "MIC" jack or equivalent.
Never try to connected to "AUX" or "LINE INPUT" jack.
Black plug
This black plug is used for the remote controlling to command start and stop of the tape recorder.
• It is not possible to control start and stop of the tape recorder from the computer unless the tape recorder has the "REMOTE" jack.
To be connected with "REMOTE" jack
Be sure to disconnect the black plug, in case the tape is to be advanced manually in such as REWIND and FAST ADVANCE.

- When a stereophonic tape recorder (or tape deck) is to be used, connect the red plug to either of jacks as it has two jacks, and connect the grey plug to the same side of the line where the red plug was connected.

MAGNETIC TAPE CONTROL INSTRUCTIONS AND TAPE RECORDER OPERATING PROCEDURES

Described in the followings are the computer and tape recorder operating procedures in recording program in the magnetic tape or reading the program from the tape recorder. Explanation will proceed assuming as though the computer has been connected with the tape recorder via the optional cassette interface.

1. Adjustments of the tape recorder and hints during adjustments

(1) For those tape recorder having the tape selector, set the selector to the type of the tape being used.

(2) In case a micro-cassette tape recorder is to be used and that if it should have the speed control switch, set the speed control to a faster speed.

(3) Program/data transfer and collation should be done with the same tape recorder that has been used in recording.
It may sometimes may not allow to perform program/data transfer or collation if a different tape recorder is used.

(4) For those tape recorders equipped with the mixing feature, set the mixing performance inactive for both recording and playbock.

- Be sure to have occassional cleaning of the recording head, as spoil on the recording head may cause to drop level and increase distortion.
- Recording tapes available on the market are usually satisfactory for the operation. But, avoid using such a tape that has extremely inferior freqeuncy characteristics or containing scratch and folds.

2. Recording to the magnetic tape (CSAVE and PRINT# COMMAND)

Recording of program, reserve program, and data into the magnetic tape should be done in the following sequence.
(In the case of manual operation)

Step	Operation	Note
(1)	Set the tape on the tape recorder.	Locate the unrecorded portion of the tape and decide the position where recording takes place. (Do not use the non-recorded portion at the leading edge of the tape.)
(2)	CE-121 Tape recorder Red plug . . . connect with "MIC" jack. Black plug . . . connect with "REMOTE" jack.	
(3)	Write program or data in the computer. EX: Write the program 1 described in Page 28 into the computer.	
(4)	Depress the "REC" and "PLAY" push-buttons of the tape recorder to make it ready for recording from the external microphone.	NOTE-1: For those equipped with the automatic level control, set the control to the automatic mode.
(5)	[Recording procedure] Execute the recording command. For program loading, perform ⌜CSAVE "file name" [ENTER] ⌟. For data transfer, perform ⌜PRINT # "file name" [ENTER] ⌟. EX: Recording program in the RUN mode. [C] [S] [A] [V] [E] [SHFT] [↑↑] [A] [A] [SHFT] [↑↑] [ENTER]	Before the commencement of recording, non-signal area will be prepared for a first 6 seconds with beeping tone, then recording takes place starting from the file name, to be followed by the recording of file contents. NOTE-2: If the tape recorder is equipped with the recording level control, adjust the tape recorder in such a manner that the recording level should show "O" point during the execution of recording command with beeping tone (either the meter hand pointing at "O" or the level indicator lighting at the point"O"). Then, perform recording again after changing the file name.

Step	Operation	Note
	When the recording is over, the display will turn on to exhibit the prompt sign. Rewind the tape and then check proper recording of the program or the data.	As for the collation, see Page 78. NOTE-3: Disconnect the black plug when the tape is to be fed in manual mode for REWIND and FAST ADVANCE operation.

- In case a tape recorder without "REMOTE" jack is to be used, depress the "PAUSE" switch before the execution of Step (4), so as to keep the tape recorder in the temporary halt state. Then, release the halt state by freeing the "PAUSE" switch before the depression of the [ENTER] key after entering the recording command at Step (5).

- When data is to be recorded (by PRINT# command) during the execution of program, finish the procedure reguired for the tape recorder before the execution of the program.

In case a tape recorder without "REMOTE" jack is to be used, insert the STOP command before the PRINT# command, so as to stop execution of the program.

EX:
```
    ⋮
150 : STOP : PRINT# "ABC" ; D
    ⋮
```

As the STOP command is executed, it comes to break program execution (with "BREAK AT 150" being displayed, in the case of the above example), then perform tape recorder operation at this point. After this, depress the "PAUSE" button of the tape recorder after making entry of [C] [O] [N] [T] from the keyboard to release the break state, and then depress the [ENTER] key after the tape begins to run.

Taking note of the tape counter number at the beginning of the recording will help much to find out the unrecorded portion.

3. Collation (CLOAD? command)

Step	Operation	Note
(1)	Set the tape to be collated on the tape recorder.	Through the tape counter, make sure that the record to be collated is in successive tape area.
(2)	CE-121 — Tape recorder Grey plug . . . connect with the "EAR-PHONE" or "MONI-TOR" jack. Black plug . . . connect with the "RE-MOTE" jack.	
(3)	Set the volume control to a setting within a range of intermediate and maximum position.	
(4)	Set the tone control (TONE, BASS, TREBLE) to settings within a range of intermediate and maximum positions.	TONE: Tone control BASS: Bass control TREBLE: Treble control

Step	Operation	Note
(5)	Push the "PLAY" button of the tape recorder to keep the tape recorder in the reproduction mode.	
(6)	[Collation] Execute the CLOAD? command, ⌜CLOAD? "file name" [ENTER]⌟. EX: [C] [L] [O] [A] [D] [SHFT] [?] [SHFT] [↑↑] [A] [A] [SHFT] [↑↑] [ENTER] When both records come to match each other, the prompt sign is exhibited on the display after the termination of collation.	Enter the file name to be collated.
	In case an error (Error-5) is encountered during the collation, change controls at Steps (3) and (4) in a slight degree, then execute the collation again to choose proper setting of the controls.	In some of tape recorders, it sometimes shows better results when the red plug is disconnected from the "MIC" jack.

If a good results is not achieved after repetition of the above procedure, put another tape recorder into use, because the present tape recorder is not properly matching in such as input/output levels, impedance, distortion factor, and phase.

- In case a tape recorder without "REMOTE" jack is to be used, depress the "PAUSE" button before the operation of Step (5) to make the tape recorder temporarily halted. Then, release the halt state by depressing the "PAUSE" button again immediately after the depression of the [ENTER] key after entering the "CLOAD?" command at Step (6).

4. Transfer from the tape (CLOAD, CHAIN, and INPUT# commands)

Observe the following procedure to transfer program, reserve program, or data from the tape to the computer.
(In the case of manual operation)

Step	Operation	Note
(1)	Set the tape to be transfered on the tape recorder.	Through the tape counter, make sure that the record to be transfered is in successive tape area.
(2)	CE-121 Tape recorder Grey plug . . . connect with "EAR-PHONE" or "MONITOR" jack. Black plug . . connect with "REMOTE" jack.	

Step	Operation	Note
(3)	Set the volume control to a setting within a range of intermediate and maximum position.	
(4)	Set the tone controls (TONE, BASS, TREBLE) to setting within a range of intermediate and maximum positions.	TONE: Tone control BASS: Bass control TREBLE: Treble control
(5)	Depress the "PLAY" button of the tape recorder to place the tape recorder in the reproduction mode.	
(6)	[Transfer operation] Execute the transfer command In the case of program, execute ⌜CLOAD "file name" [ENTER]⌟. In the case of data, execute ⌜INPUT "file name" [ENTER]⌟. Ex: [C] [L] [O] [A] [D] [SHFT] ["] [A] [A] [SHFT] ["] [ENTER] After the termination of transfer, the prompt sign is exhibited on the display.	Enter the file name that has been recorded.
	If error status (Error-5) is encountered, change controls at Steps (3) and (4) in a slight degree, then execute the transfer again to choose proper setting of controls.	In some of tape recorders, it sometimes shows better transfer results when the red plug is disconnected form the "MIC" jack.

- In case a tape recorder without "REMOTE" jack is to be used, depress the "PAUSE" button before the operation of Step (5) to make the tape recorder temporarily halted. Then, release the halt state by depressing the "PAUSE" button again immediately after the depression of the [ENTER] key upon entering the transfer command at Step (6).
- If the transfer is to be carried out during the execution of program (CHAIN and INPUT# command), perform the necessary tape recorder operation before the execution of the program.

⌜In case a tape recorder without "REMOTE" jack is to be used, insert the STOP command before the CHAIN or INPUT# command to stop execution of program, and perform necessary tape recorder operation, as in the recording.⌟

[Other precautions]

(1) Connect the computer securely with the CE-121 and do not pull the plugs out during the operation. Be sure to turn the power off on the computer first prior the connection/disconnection of the CE-121.

(2) When connected operation results in failure in operating the system with ac power source or ac adaptor or that if the computer does not operate in a normal condition, operate the system from the battery power.
Using the tape recorder by means of ac power under an environment where much noise is prevailing on the ac line, it may cause to interrupt normal computer operation or joint operation. In such a case, use of a noise suppression device sold on the market may work to prevent noise involvement.

FUNCTIONS OF KEYS

Below are explained the functions of main keys.

Key	Function
CA/BREAK [ON]	• Used to power on. • Breaking (temporarily interruptting) the program being executed. • Clearing the computer completely. (Reset of error condition.)
[OFF]	• Pressed to power off.
[SHFT]	• Designating instructions (functions) which are not given on the keys but described on the penel in mustard, such as π and $\wedge$ symbols. [2nd function designation] Ex. [SHFT] [$\overset{\pi}{1}$] $\rightarrow \pi$ input • In "DEF" mode, pressed prior to keying in the label to execute a program defined by it, such as A, S, D, etc. (Definable key designation) . . . . . Ex. [SHFT] [A] • In "RESERVE" mode, pressed prior to pushing the reserved or reserving key when recalling or reading in a reserve program. (Reserve key designation) . . . . . Ex. [SHFT] [B] • In "PRO" or "RUN" mode, pressed prior to pushing the reserved key when recalling a reserve program. . . . . Ex. [SHFT] [B] (Reserve key designation)
[0] ~ [9]	• Used to enter numbers.
[.]	• Specifying the position of a decimal point. • Designating abbreviations when inputting instructions. • Designating to display numbers following a decimal point in numerical data by a USING statement instruction. (See page 58.)
[Exp]	• Designating to input exponents. (The symbol of this key is displayed as E.)
[SHFT] [π]	• Designating Pi (π)
[A] ~ [Z]	• These alphabetical keys serve to designate instructions. • Specifying variables (A to Z memory)
[/]	• Designating division instructions.
[*]	• Designating multiplication instructions.
[+]	• Displaying a plus sign for numbers. (Usually omitted.) • Designating addition instructions.
[−]	• Displaying a negative sign for numbers. • Designating substraction instructions.
[SHFT] [$\wedge$]	• Designating power calculation instructions. • Specifying the floating decimal point system (exponent display) for numerical data in USING statement instructions.
[SHFT] [<]	• Specifying logical operators, such as <, <=, <>.

Key	Function
SHFT >	• Specifying logical operators, such as >, >=, <>.
=	• In assignment statements, designating to assign the content (number or character) on the right side for the variable specified on the left side. • Specifying logical operators, such as =, <=, >=.
(,)	• Designating parentheses.
√	• Designating extraction of the square root.
SPC	• Instructing to provide space when inputting programs or characters. The space is not considered in programming, executing operations, etc.
SHFT :	• Designating pauses in multi-statement (in which two or more statements are defined in one line).
SHFT ;	• In PRINT statement instructions, designating multi-display (by which two or more contents are displayed at a time). • In INPUT statement instructions, designating pauses in comment. • In PRINT # statement and INPUT # statement instructions, designating pause between the instruction and the variable.
SHFT ,	• Designating pause between two equations in continuous calculation. • In PRINT statement instructions, designating double display (in which two different contents are displayed at a time). • In INPUT statement instructions, specifying pause between comments or variables. • In CHAIN statement, specifying pause between file and expression, or between file and label when setting the opening line subsequent to execution.
SHFT #	• In USING statement, specifying the instruction to define the display format of numerical data. • Designating PRINT # and INPUT # statements.
SHFT ?	• Designating CLOAD? statement.
SHFT $	• Designating character variables.
SHFT "	• Designating and cancelling characters. • Specifying labels.
MODE	• Changing modes (DEF, RUN, PRO, RESERVE).
CL	• Executing instructions given to eliminate the contents input by manual operation. • Executing instructions given to clear the displayed contents such as calculation results. • Resetting errors.
►	• Shifting the cursor to the right. • Executing playback instructions. • Recalling the cursor in case it is not displayed in appearing programs or reserve programs. (next to colon's right)

Key	Function
◄	• Shifting the cursor to the left. • In other functions, the same as the ► key.
SHFT INS	• Executing an instruction given to make an insertion space (⊏ appears, however) of 1-step capacity between the address (N) indicated by the cursor and the preceding address (N−1).
SHFT DEL	• Executing an instruction given to delete the contents of the address (N) indicated by the cursor.
ENTER	• Designating an instruction for completing the program line. • Writing in programs or reserve programs. • Executing manual calculation or directing execution of a COMMAND statements. • Executing a restart instruction after inputting data in waiting for data given by INPUT statement or after executing PRINT statement.

The ↑ , ↓ and ON keys have the following functions, depending on designated modes as well as the state of the computer.

<table>
<tr><th>Mode</th><th>State</th><th>↑</th><th>↓</th><th>ON</th></tr>
<tr><td></td><td>Power off</td><td></td><td></td><td>To power on</td></tr>
<tr><td rowspan="4">RUN

DEF</td><td>Program being executed</td><td></td><td></td><td rowspan="3">To bring into BREAK (program is temporarily interrupted.)</td></tr>
<tr><td>INPUT statement being executed</td><td rowspan="3">To display program line being executed or already executed, by keeping this key depressed.</td><td rowspan="2">To execute debugging operation</td></tr>
<tr><td>PRINT statement just now executed</td></tr>
<tr><td>Under BREAK</td><td>To execute the next line</td><td rowspan="2">To clear completely</td></tr>
<tr><td>Error condition during executing program</td><td>To display error making line, by keeping this key depressed</td><td></td></tr>
<tr><td colspan="5">PRO (In the case when program line is not being displayed; e.g. such case as when changing any other mode to PRO mode)</td></tr>
<tr><td></td><td>PRINT statement just now executed</td><td rowspan="2">To display the interrupted line</td><td rowspan="2">Same as left</td><td></td></tr>
<tr><td></td><td>Under BREAK</td><td></td></tr>
<tr><td></td><td>Error has been cleared with any key other than the ON key</td><td>To display program line in which the error occurred</td><td>Same as left</td><td></td></tr>
<tr><td></td><td>Others</td><td>To display the end line</td><td>To display the head line</td><td></td></tr>
</table>

Mode	State	↑	↓	ON
PRO (In the case when program line is being displayed)		To display the preceding program line	To display the next program line	To clear completely
RESERVE				

- Excepting the ENTER, ON, OFF, ↑, ↓, ►, ◄, CL, INS, DEL, MODE, SHFT and " keys, the symbols of all the other keys that are in quotations (" ") are defined to be characters.
  (!, %, or ¥ may be defined only as a character.)
- When the A, S, D, F, G, H, J, K, L, =, Z, X, C, V, B, N, M, or SPEC keys has been depressed, following the SHFT key operaters.
  1) In DEF mode, a program defined with the label of the same character begins being executed.
  2) In RESERVE mode, a reserve program is recalled or written in.
  3) In PRO or RUN mode, the contents reserved by the key is recalled. If nothing is reserved, the symbol of the key is displayed.
- The ENTER key provides a space on the display.
- The OFF key does not function when operations are being done in the machine, such as during execution of programs.
- In keying-in wait condition, the power is automatically turned off if any key has not been pushed for about 7 minutes. It depends on operating conditions, etc. (Auto power off)

BATTERY REPLACEMENT

When the battery indicator is out, replace the designated mercury batteries*.

1. Turn off the computer.
2. Remove the screws from the back cover with a small screw driver (Fig. 1).
   (Please note that two kinds of screw are used.)
3. Replace the batteries. (Fig. 2) (See note (1))
4. Hook the tabs of the back cover into the slits of the computer proper. (Fig. 3)
5. Push the back cover in slightly while replacing the screws.
   Tighten the screws securely at this time.
6. Push the reset switch on the back cover to clear the computer. (Fig. 4)
   Use a ball-point pen to press the reset switch.
7. Press the [OFF] and [ON] keys to clear the computer. When the batteries are correctly installed ">" DEG RUN "." will be displayed.

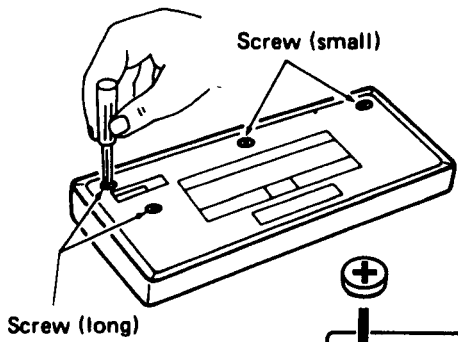


Fig. 1

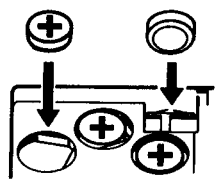
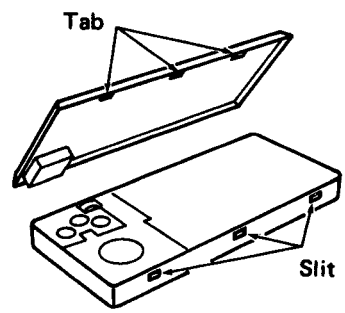


Fig. 2



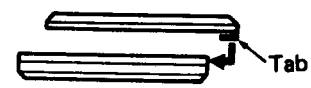


Fig. 3

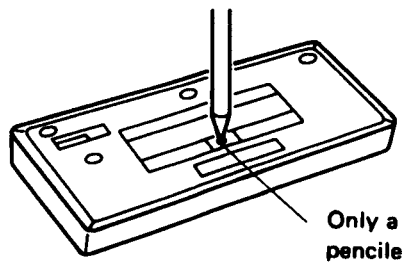
Note (1) : When replacing the batteries, observe the following instructions to prevent the failure of the set due to improper battery replacement.

- Always replace all 4 batteries at the same time.
- Do not mix new batteries with used batteries.
- Do not use different kinds of batteries together. (This computer cannot be operated on silver oxide battery nor alkaline manganese battery, though the shape is quite similar to the mercury battery.)
- Wipe off the surface of the new batteries with dry cloth and then, install the batteries as shown in Fig. 2.

* Battery

- **Mercury battery (Type MR44, 1.35V) x 4**

Batteries may be obtained where you purchased your computer or at most retail outlets for calculators, watches, or cameras.



Only a little pressure is needed. Do not use a pencile or other materials that could break in the depressions.

Fig. 4

Note (2) : The uppermost line of the dot matrix display may light up in all of 24 digits for about one second when the machine is energized by depressing the [ON] key, or the reset switch. This indicates none of troubles, however.

SPECIFICATIONS

Model:	PC-1211	
Number of calculation digits:	10 digits (mantissa) + 2 digits (exponent)	
Calculation system:	According to mathematical formula (with priority judging function)	
Program system:	Stored system	
Program language:	BASIC	
Capacity:	Program memory;	Max. 1424 steps
	Data memory;	Fixed memory . . . 26 pcs. Flexible memory (common with program memory) . . . Max. 178 pcs.
	Reserve memory;	Max. 48 steps (reserve program: Max. 18 kinds)
	Input buffer;	80 characters
Stack:	For data;	8 steps
	For function;	16 steps (in parentheses, 15 levels)
	For subroutine;	4 steps
	For FOR-NEXT statement;	4 steps
Calculations:	Four arithmetic calculations, power calculation, trigonometric and inverse trigonometric functions, logarithmic and exponential functions, angular conversion, extraction of square root, sign function, absolutes, integers, and logical calculations.	
Editing function:	Cursor shifting (►, ◄) Insertion (INS) Deletion (DEL) Line up and down (↑, ↓)	
External memory function:	By using the optionally available cassette interface 〈CE-121〉, program, reserve program, and data memory can be recorded or read out to or from magnetic tape (tape recorder).	
Memory protection:	CMOS battery back-up (program, data and reserve memories are protected)	
Display:	24-digit alphanumeric dot matrix liquid crystal display	
Component:	CMOS LSI, etc.	
Power supply:	5.4 V ⎓ (DC): Mercury battery (1.35V) x 4	
Power consumption:	5.4 V ⎓ (DC): 0.011W 5.4 V ⎓ (DC): 0.013W (with CE-121)	
Operating time:	Approx. 300 hours (Type MR-44) at the condition of the display 5 5 5 5 and the ambient temperature 20°C (68°F). └Cursor The operating time slightly changes depending on the type of battery or the way of use.	
Operating temperature:	0°C ~ 40°C (32°F ~ 104°F)	
Dimensions:	175(W) x 70(D) x 15(H) mm 6-7/8"(W) x 2-3/4"(D) x 19/32"(H)	
Weight:	Approx. 170g (0.37 lbs.)	
Accessories:	Carrying case, Mercury battery (Type MR-44) x 4 (Built-in)	

LIST OF FUNCTIONS AND STATEMENTS

Below are compiled the functions and statements for your reference.
The abbreviations input in this machine will be displayed as the individual basic forms when being converted into an instruction.

1. Functions

Remember to press the [•] key.

Functions	Abbreviations	Remarks	Ref. page
SIN	SI.	sin — Trigonometric functions	18
COS		cos — Trigonometric functions	
TAN	TA.	tan — Trigonometric functions	
ASN	AS.	$\sin^{-1}$ — Inverse trigonometric functions	18
ACS	AC.	$\cos^{-1}$ — Inverse trigonometric functions	
ATN	AT.	$\tan^{-1}$ — Inverse trigonometric functions	
LN		$\log_e X$ Natural logarithm — Logarithmic functions	19
LOG	LO.	$\log_{10} X$ Common logarithm — Logarithmic functions	
EXP	EX.	e^x Exponential function (Antilogarithm for LN)	19
$\sqrt{\ }$		Extraction of square root	19
DMS	DM.	Decimal to degree/minute/second conversion	19
DEG		Degree/minute/second to decimal conversion	19
INT		Integer	20
ABS	AB.	Absolute value	20
SGN	SG.	Signum	20

2. Statements

Statements	Abbreviations	General forms	Remarks	Ref. page
LET (assignment statement)	LE.	(1) LET [numerical variable] = ⟨ expression ⟩ (2) LET [Character variable] = "character" (3) LET [Character variable] = [Character variable]	LET can be omitted (except the case in which IF statement is followed by it).	52
INPUT	I. IN. INP. INPU.	(1) INPUT [variable], [variable], ··· (2) INPUT "character", [variable], "character", [variable], ··· (3) INPUT "character"; [variable], "character"; [variable], ···	Input instruction Data is input.	53
PRINT	P. PR. PRI. PRIN.	(1) PRINT ⟨ expression ⟩ (2) PRINT "character" (3) PRINT [Character variable] (4) PRINT { ⟨ expression ⟩ / "character" / [Character variable] }, { ⟨ expression ⟩ / "character" / [Character variable] } (5) PRINT { ⟨ expression ⟩ / "character" / [Character variable] }; { "character" / [Character variable] }; ··· { "character" / [Character variable] }	Output instruction Specified contents are displayed.	55
PAUSE	PA. PAU. PAUS.	General froms are the same as those for PRINT statement.	Output instruction Specified contents are programmed after being displayed for about 0.85 second.	58
USING	U. US. USI. USIN.	(1) USING "#...#.#...#∧" (2) USING (end of statement) ENTER or : (colon) (3) (a) { PRINT / PAUSE } USING "FORMAT", ··· (b) { PRINT / PAUSE } USING; ···	Format designation instruction Displaying format for numerical data is designated. Format designation is cancelled.	58
GOTO	G. GO. GOT.	(1) GOTO ⟨ expression ⟩ (2) GOTO { "character" / [Character variable] }	Jump instruction Specified line or label is executed.	60
IF		(1) IF ⟨ expression ⟩ logic operator ⟨ expression ⟩ execution statement (2) IF ⟨ expression ⟩ excution statement (3) IF { "character" / [Character variable] } = { "character" / [Character variable] } execution statement (4) IF [Character variable] execute statement	Decision instruction Given conditions are decided enabling to shift the execution to the next execute statement or the next line.	61
THEN	T. TH. THE.	This statement is defined as a execution statement in an IF statement. General form is the same with that of GOTO statement.	Jump instruction Possible only to define as execute statement in IF statement.	62

State-ments	Abbrevia-tions	General forms	Remarks	Ref. page
GOSUB	GOS. GOSU.	(1) GOSUB ⟨ expression ⟩ (2) GOSUB { "character" / [Character variable] }	Subroutine jump instruction The execution is shifted to specified line or label, where the program is executed as sub-routine.	63
RETURN	RE. RET. RETU. RETUR.	RETURN	Return instruction By this subroutine return instruction, the execution is returned to the statement following the GOSUB state-ment.	63
FOR STEP	F. FO. STE.	(1) FOR [numerical variable] = ⟨ expression 1 ⟩ TO ⟨ expression 2 ⟩ (2) FOR [numerical variable] = ⟨ expression 1 ⟩ TO ⟨ expression 2 ⟩ STEP ⟨ expression 3 ⟩ ⟨ expression 1 ⟩: Initial value ⟨ expression 2 ⟩: End value ⟨ expression 3 ⟩: Increment	FOR loop start Used in combina-tion with NEXT statement.	65
NEXT	N. NE. NEX.	NEXT [numerical variable] This [numerical variable] must correspond to that for FOR statement.	FOR loop end is indicated. Used in combination with FOR statement.	65
STOP	S. ST. STO.	STOP	To stop executing program.	69
END	E. EN.	END	To indicate program end.	69
BEEP	B. BE. BEE.	BEEP ⟨ expression ⟩	Beep sound instruc-tion Beep tone is generat-ed as many times as the number of values in ⟨ expression ⟩.	69
CLEAR	CL. CLE. CLEA.	CLEAR (Possible to execute by manual operation) CLEAR [ENTER]	Data memory clear instruction	69
DEGREE	DEG. DEGR. DEGRE.	DEGREE (Possible to execute by manual operation) DEGREE [ENTER]	Angurar mode designation Degree (°) is designated.	70
RADIAN	RA. RAD. RADI. RADIA.	RADIAN (Possible to execute by manual operation) RADIAN [ENTER]	Angurar mode designation Radian ([rad]) is designated.	70

State-ments	Abbrevia-tions	General forms	Remarks	Ref. page
GRAD	GR. GRA.	GRAD (Possible to execute by manual operation) GRAD [ENTER]	Angurar mode designation Grad ([g]) is designated.	70
AREAD (auto read)	A. AR. ARE. AREA.	AREAD [variable]	The contents displayed at program definable start is memorized in the specified [variable].	70
REM (remark)		REM ⟨ note ⟩	To designate non-execute statement (note) in program.	71
⟨ **Command statement** ⟩ Possible only to execute by manual operation.				
RUN	R. RU.	(1) RUN [ENTER] (2) RUN ⟨ expression ⟩ [ENTER] (3) RUN { "character" / [Character variable] } [ENTER]	Program execute start instruction Effective only in DEF and RUN modes.	71
DEBUG	D. DE. DEB. DEBU.	The general forms are defined in the same manner as those for RUN statement.	Debugging start instruction Effective only in DEF and RUN modes.	72
CONT	C. CO. CON.	CONT [ENTER]	To restart an interrupted program Effective in DEF and RUN modes.	72
LIST	L. LI. LIS.	The general forms are defined in the same manner as those for RUN statement.	For listing programs Effective in PRO mode.	73
NEW		NEW [ENTER]	In DEF, RUN and PRO modes, program memory and data memory are completely cleared. In RESERVE mode, reserve memory is cleared.	74
MEM	M. ME.	MEM [ENTER]	Remaining area of a program memory is displayed with the number of program steps and that of flexible memories.	74

Statements	Abbreviations	General forms	Remarks	Ref. page
⟨ Magnetic tape control statement ⟩				
CSAVE (cassette save)	CS. CSA. CSAV.	CSAVE "file name" [ENTER] (Possible only by manual operation)	Program or reserve program is recorded in magnetic tape.	77
CLOAD (cassette load)	CLO. CLOA.	CLOAD "file name" [ENTER] (Possible only by manual operation)	Program or reserve program is transferred from magnetic tape to the computer.	78
CLOAD? (cassette load?)	CLO. ? CLOA. ?	CLOAD? "file name" [ENTER] (Possible only by manual operation)	The contents of program or reserve program and those of magnetic tape are checked up.	78
CHAIN	CH. CHA. CHAI.	(1) CHAIN "file name" (2) CHAIN "file name", ⟨ expression ⟩ (3) CHAIN "file name" { "character" / [Character variable] } (To be executed by program)	Program recorded in magnetic tape is read in and then executed.	79
PRINT #	P. # PR. # PRI. # PRIN. #	(1) PRINT # "file name" (2) PRINT # "file name"; [Label of variable] (Possible to execute both by program and manual operation)	Data memory contents are recorded in magnetic tape.	81
INPUT #	I. # IN. # INP. # INPU. #	(1) INPUT # "file name" (2) INPUT # "file name" ; [Label of variable] (Possible to execute both by program and manual operation)	Data recorded in magnetic tape is transferred in data memory of the computer.	82

<CALCULATION RANGE>

Functions	Dynamic range
$y \wedge x$ (y^x)	$0 \leqq y$, $-1 \times 10^{100} < x \log y < 100$ (Here, $y \wedge x = 0$ at $y = 0$)
SIN x COS x TAN x	DEG: $\lvert x \rvert < 1 \times 10^{10}$ RAD: $\lvert x \rvert < \frac{\pi}{180} \times 10^{10}$ GRAD: $\lvert x \rvert < \frac{10}{9} \times 10^{10}$ In TAN x, however, the following cases are excluded. DEG: $\lvert x \rvert = 90\ (2n - 1)$ RAD: $\lvert x \rvert = \frac{\pi}{2}\ (2n - 1)$ GRAD: $\lvert x \rvert = 100\ (2n - 1)$ (n = integer)
$\text{SIN}^{-1}\ x$ $\text{COS}^{-1}\ x$	$-1 \leqq x \leqq 1$
$\text{TAN}^{-1}\ x$	$\lvert x \rvert < 1 \times 10^{100}$
LN x LOG x	$1 \times 10^{-99} \leqq x < 1 \times 10^{100}$
EXP x	$-1 \times 10^{100} < x \leqq 230.2585092$
$\sqrt{x}$	$0 \leqq x < 1 \times 10^{100}$

Functions other than shown above can be calculated only when x stays within the following range.

$1 \times 10^{-99} \leqq \lvert x \rvert < 1 \times 10^{100}$ and 0

– MEMO –